How Algebra Is Born from Cellular Automata: Algebraic Abiogenesis, Scale-Dependent Symmetry, Contextual Monoids, and Emergent Mathematical Laws

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

July 13, 2026

Can algebraic structures arise from simple dynamics without being specified in advance? This paper develops Cellular Algebraic Abiogenesis as a formal and computational framework for studying how objects, operations, symmetries, and laws first become available under changes of scale, context, quotienting, and observation policy. Elementary cellular automata provide a controlled substrate: their microscopic rules are explicit, finite, and local, yet their block dynamics can support exact higher-level algebras absent from the corresponding microscopic term structure. We present four complementary phenomena. Rule 178 exhibits a scale-three semilattice quotient whose commutative, associative, and idempotent operation is not inherited from any microscopic binary minor. Rule 30 generates the full transformation monoid on two-bit blocks, including exact context-controlled synthesis of NAND. Rules 60 and 90 produce arithmetic towers of affine and linear symmetry groups governed by the odd part of the observation scale and Frobenius decomposition. Across all elementary rules, symmetries can be born, persist, or disappear between scales. These results recast Algebraic Abiogenesis as the transition from latent operational capacity to explicit, policy-indexed algebraic organization, rather than mathematics arising from literal nothingness.

Keywords: Algebraic Abiogenesis, cellular automata, emergent algebra, scale-dependent symmetry, contextual transformation monoids, universal algebra, block dynamics, coarse-graining, quotient algebra, semilattice, Rule 178, Rule 30, Rule 60, Rule 90, transformation monoid, Frobenius decomposition, affine groups, algebraic emergence, observation policy, mathematical discovery.

A collaboration with OpenAI GPT-5.5 Thinking. CC4.0. 125 pages and 63 references.

https://orcid.org/0009-0004-4206-4253

Code: https://github.com/DougYouvan/cellular-algebraic-abiogenesis

Outline

1. From Anonymous Dynamics to Algebraic Organization

1.1 The problem of deriving mathematics from lower-level operations

1.2 Why operation tables begin too high in the algebraic hierarchy

1.3 Cellular automata as pre-algebraic experimental substrates

1.4 From mathematics ex nihilo to organizational emergence

1.5 Central claims and contributions of the paper

2. Algebraic Abiogenesis: Definitions and Criteria

2.1 Algebraic objects, operations, transformations, and laws

2.2 The substrate-policy-algebra relation

2.3 Strong Algebraic Abiogenesis

2.4 Weak Algebraic Abiogenesis and algebraic morphogenesis

2.5 Algebraic revelation and inherited structure

2.6 Exactness, noninheritance, threshold birth, and certification

2.7 Algebraic genesis, persistence, and extinction

3. Cellular Automata as a Laboratory for Mathematical Emergence

3.1 Elementary cellular automata and anonymous local rules

3.2 Microscopic states, neighborhoods, and spacetime evolution

3.3 Exact block dynamics and supercell operations

3.4 Observation scale as part of the mathematical experiment

3.5 Context selection, quotienting, and admissibility policies

3.6 Why a cellular automaton supports an ecology rather than one algebra

4. Exact Block Algebras and Microscopic Noninheritance

4.1 Construction of the scale-k block algebra

4.2 Central-block evolution as a ternary operation

4.3 Congruences and exact quotient formation

4.4 Binary minors and term-derived operations

4.5 Testing whether a macro-law already exists microscopically

4.6 Complete congruence searches over elementary cellular automata

4.7 Certificates of genuine algebraic law birth

5. Rule 178 and the Threshold Birth of a Semilattice

5.1 The microscopic asymmetry of Rule 178

5.2 The homogeneous-mixed-homogeneous quotient policy

5.3 Failure at block sizes one and two

5.4 Exact formation of the quotient at block size three

5.5 The emergent commutative, associative, idempotent operation

5.6 Semilattice structure and its order-theoretic interpretation

5.7 Domain-wall dynamics and the Rule 222 representation

5.8 Proof of persistence at every block size greater than or equal to three

5.9 Why the semilattice is not inherited from microscopic binary minors

6. Contextual Transformation Monoids

6.1 From algebra on states to algebra on transformations

6.2 One-variable contexts of exact supercell operations

6.3 Definition of the contextual transformation monoid

6.4 Composition as the emergent algebraic operation

6.5 Units, singular maps, ranks, and transformation structure

6.6 Contextual monoids as scale-indexed invariants

6.7 Relation to translation monoids and universal algebra

6.8 Controlled context composition versus autonomous CA evolution

7. Rule 30, Full Transformation Completeness, and NAND

7.1 The two-cell block state space

7.2 Exact generation of the full transformation monoid T_4

7.3 A minimal three-context generating certificate

7.4 Emergence of the non-Abelian symmetric group S_4

7.5 Exact two-context synthesis of embedded NAND

7.6 Functional completeness at the macrostate level

7.7 The fourteen elementary rules generating T_4

7.8 Symmetry classes and equivalence among the fourteen rules

7.9 What contextual functional completeness does and does not imply

8. Contextual Symmetry Spectra Across Scale

8.1 The unit group of a contextual transformation monoid

8.2 Definition of the contextual symmetry spectrum

8.3 Complete elementary-rule classification at scales two and three

8.4 Symmetry genesis between adjacent scales

8.5 Persistence of non-Abelian organization

8.6 Symmetry extinction under increased block size

8.7 The scale-three bifurcation of the T_4 rules

8.8 Failure of simple monotonicity in algebraic complexity

8.9 Algebraic universality classes based on scale spectra

9. Frobenius Towers and Arithmetic Selection of Symmetry

9.1 Additive cellular automata over finite fields

9.2 Laurent-polynomial representations of local dynamics

9.3 Frobenius decomposition under characteristic-p scale multiplication

9.4 Residue-class splitting of supercell coefficient matrices

9.5 Contextual-group towers along characteristic-primary scale rays

9.6 Preservation of the nontranslation symmetry core

9.7 Translation-module inflation under scale multiplication

9.8 Arithmetic observation scales as selectors of emergent algebra

10. Rule 60 and Rule 90 as Distinct Algebraic Phase Diagrams

10.1 The odd part of the observation scale

10.2 The Rule 60 affine-group formula

10.3 Abelian behavior at power-of-two scales

10.4 Full affine symmetry at odd scales

10.5 Solvability and non-Abelian phase transitions

10.6 The Rule 90 Jordan-pair decomposition

10.7 Product symmetry cores and exceptional dimensions

10.8 Comparison of Rule 60 and Rule 90

10.9 Weak Algebraic Abiogenesis in additive cellular automata

11. General Theory, Novelty, and Philosophical Limits

11.1 Algebraic ecologies indexed by observation policy

11.2 Scale as an algebraically causal variable

11.3 Latent operational capacity and explicit mathematical organization

11.4 Comparison with coarse-graining, renormalization, and emergent computation

11.5 Comparison with universal algebra and translation monoids

11.6 What is established prior art

11.7 What appears novel in the present framework

11.8 Why cellular automata are not nonmathematical substrates

11.9 The limits of the phrase mathematics from nothing

11.10 Algebraic Abiogenesis as a precise form of mathematical emergence

12. Computational Reproducibility, Future Research, and Conclusion

12.1 Exhaustive searches, exact verification, and machine-readable certificates

12.2 The public computational repository

12.3 Reproduction standards and falsifiability

12.4 Extension to larger neighborhoods and multistate cellular automata

12.5 Automated discovery of signatures beyond groups and semilattices

12.6 Categories, groupoids, operads, and process algebras

12.7 Cross-substrate recurrence of identical emergent theories

12.8 Policy optimization and autonomous selection of mathematical objects

12.9 Implications for no-corpus mathematical artificial intelligence

12.10 Toward a general science of algebraic birth

12.11 Final conclusions

13. References

1. From Anonymous Dynamics to Algebraic Organization

1.1 The problem of deriving mathematics from lower-level operations

Traditional mathematics begins with named objects and named operations. A group is introduced together with a binary operation, a ring with two operations, a lattice with meet and join, and a category with objects and morphisms. The mathematical task then consists of proving consequences from a supplied signature and a supplied set of axioms. This approach is extraordinarily powerful, but it leaves one foundational question largely untouched: where do the objects, operations, and axioms themselves come from?

Algebraic Abiogenesis begins from the opposite direction. Instead of assuming that the relevant mathematical structure is already known, it asks whether a lower-level operational substrate can generate a higher-level algebraic organization that was not explicitly specified in advance. The substrate may consist of anonymous finite operations, local transition rules, spacetime evolution, or other deterministic mechanisms. The discovery system is not told to find a group, a semilattice, an affine space, or a transformation monoid. It must first determine what distinctions are stable, what combinations are operationally meaningful, and which exact laws are supported by the resulting objects.

The central problem is therefore not ordinary theorem proving. In theorem proving, the language of the theory is already available. In Algebraic Abiogenesis, the mathematical language itself is part of the unknown. The task includes the formation of operands, the extraction of operations, the discovery of admissible equivalences, and the certification of relations among composite expressions.

This shifts the guiding question from

What follows from these axioms?

to

Which algebraic theory becomes available from these dynamics?

The distinction is fundamental. A theorem prover searches inside a pre-existing formal world. An abiogenic system attempts to construct the formal world in which the theorem can later be stated.

The phrase “lower-level operations” must be interpreted carefully. No computer program or cellular automaton is devoid of mathematics. A finite state set, a neighborhood relation, an update rule, and discrete time are already mathematical structures. The claim is not that

algebra emerges from literal non-mathematics. The claim is that a higher algebraic organization can arise from a substrate that does not explicitly contain that organization as a named theory, a supplied signature, or a target law.

The analogy with biological abiogenesis is therefore organizational rather than metaphysical. Life did not emerge from matter without chemistry. It emerged when chemical processes became organized into persistent, reproducing, regulated systems. Likewise, Algebraic Abiogenesis concerns the transition from local operations to stable mathematical objects, from interactions to composable transformations, and from repeated behavior to explicit algebraic law.

1.2 Why operation tables begin too high in the algebraic hierarchy

The earliest form of the present program began with anonymous finite operation tables. A finite set was supplied together with an operation table whose entries had no mathematical labels. Terms were generated, evaluated, and compared. When two terms induced the same function on all assignments, the equality between them became a candidate law. This approach can discover equational regularities without naming associativity, commutativity, identity, or inverse in advance.

That method remains valuable, but it begins higher in the hierarchy than its initial language suggests. An anonymous binary table already presupposes a considerable amount of algebraic organization. It assumes that there are stable objects, that two inputs can be selected, that their order is meaningful, that the operation produces one output, and that the result remains inside the same set. In other words, it assumes arity, closure, operand identity, and composability before discovery begins.

A binary operation table has the form

$$\mu: X \times X \to X.$$

This compact notation already contains several commitments. The set X has been isolated. Equality on X has been fixed. The product $X \times X$ determines how inputs are grouped. The function μ is total and single-valued. The codomain is the same as the domain of the operands. All of these features are characteristic of a developed algebraic representation.

For many natural systems, those commitments are not initially justified. Interactions may be partial, phase-dependent, multivalued after coarse observation, or capable of producing several outputs. Objects may persist only relative to a background. The effective operands may not be microscopic states but extended patterns, transformations, or equivalence classes. The correct

algebraic signature may contain ternary operations, partial compositions, typed morphisms, or entire families of context-dependent maps.

If the goal is to study the birth of algebraic organization, an anonymous operation table may therefore begin after several of the most important birth events have already occurred. It can discover laws of an operation, but it cannot explain how the operation became identifiable as an operation in the first place.

Cellular automata offer a lower starting point. Their microscopic rule specifies only local state transition. It does not explicitly specify particles, block objects, collision operations, quotient algebras, semilattices, transformation monoids, or symmetry groups. These higher structures must be extracted through observation, aggregation, and composition. The cellular-automaton program therefore moves the emergence boundary downward from anonymous algebra to anonymous dynamics.

1.3 Cellular automata as pre-algebraic experimental substrates

A cellular automaton consists of a discrete lattice, a finite set of cell states, a local neighborhood, and a deterministic update rule. In an elementary one-dimensional binary cellular automaton, each cell takes a value in $\{0, 1\}$, and the next value of a cell depends on the current values of that cell and its two nearest neighbors.

The microscopic evolution can be written as

$$x_i(t+1) = f(x_{i-1}(t), x_i(t), x_{i+1}(t)),$$

where $f: \{0,1\}^3 \to \{0,1\}$is the local rule.

This description is mathematically explicit, but it is pre-algebraic relative to the higher structures sought in this paper. The rule does not announce that a particular block should count as an object. It does not state which environmental arrangements should count as contexts. It does not identify a quotient relation or name a macro-operation. It does not tell the observer that a semilattice, a full transformation monoid, or a general linear group will appear at some block scale.

Cellular automata are especially useful because the microscopic substrate is completely known. There is no uncertainty about the local law, no measurement noise, and no inaccessible hidden mechanism. Any higher structure can therefore be traced back exactly to local evolution. At the same time, the space of possible block dynamics grows rapidly, allowing complex algebraic organization to arise from extremely small rules.

A block of $k$adjacent cells can be treated as a macrostate in

$$B_k = \{0,1\}^k.$$

Three adjacent k-cell blocks may be evolved for $k$time steps, after which the central k-cell block is read. This defines an exact supercell operation

$$\Phi_{F,k}: B_k^3 \to B_k.$$

Nothing approximate has occurred. The macro-operation is derived exactly from the microscopic rule. Once $\Phi_{F,k}$is available, several kinds of algebraic analysis become possible.

One may search for congruences of the block algebra and construct exact quotients. One may fix two arguments and vary the third, obtaining unary context maps. The maps generated by composition form a transformation monoid. The invertible maps form a permutation group. Binary terms derived from the ternary operation may satisfy identities absent from microscopic term operations. Across different values of k, these structures form scale-dependent spectra.

The CA therefore functions as a controlled experimental universe in which the following transition can be studied:

local rule ⟶ block dynamics ⟶ formed operations ⟶ algebraic laws.

The use of the term “pre-algebraic” is relative, not absolute. The CA is already a mathematical dynamical system. It is pre-algebraic only with respect to the particular higher-level algebra that has not yet been selected, formed, or named.

1.4 From mathematics ex nihilo to organizational emergence

The initial ambition of Algebraic Abiogenesis was sometimes expressed as the discovery of mathematics from nothing. Taken literally, that phrase cannot be defended. A computer executing a search already embodies logic, arithmetic, state distinction, memory, and causal sequencing. A cellular automaton already contains a finite alphabet, discrete geometry, time evolution, and a transition function. A human interpreting the output contributes additional mathematical capacities.

The stronger and more precise claim concerns organization.

A substrate can possess the operational capacity for a higher algebra without explicitly presenting that algebra as an organized object. Rule 178 does not contain a line of code stating that a three-element semilattice should appear at block scale three. Rule 30 does not explicitly list all 256 transformations of a four-state block space. Rule 60 does not name the affine groups

that occur at odd scales or describe the dependence of its contextual symmetry on the odd part of the block size.

These structures are latent in the sense that they are consequences of the rule. They are emergent in the sense that they become explicit only after a new level of object formation, contextualization, or quotienting has been introduced.

The relevant distinction is between logical implication and organized availability. Every exact macro-level result is ultimately implied by the microscopic rule. Nevertheless, the macroalgebra may not be present at the microscopic level as an accessible operation, a named object, or a law in the microscopic term language. Its formation may require aggregation over space, evolution through time, identification of states under a policy, and closure under composition.

This yields a more disciplined formulation:

> Algebraic Abiogenesis
> $=$ the transition from latent operational capacity to explicit algebraic organization.

The term “birth” therefore refers to the first policy and scale at which a structure becomes exactly available. A law may be absent at scale one, appear at scale three, and persist thereafter. A non-Abelian group may appear at one scale and disappear at the next. A transformation monoid may become functionally complete only after blocks are formed. The birth event is not the creation of truth from nothing; it is the formation of a new exact mathematical level.

This interpretation also avoids a false opposition between reduction and emergence. The macroalgebra remains derivable from the microdynamics, so it is not independent of the substrate. Yet its objects, operations, and laws may be irreducible in practice to the vocabulary used at the microscopic level. The higher theory is both grounded and novel in organization.

1.5 Central claims and contributions of the paper

This paper advances five central claims.

First, cellular automata can support exact algebraic structures that are not explicitly specified in their local rules. These structures can be discovered computationally without supplying the target law or target algebraic signature.

Second, observation policy is constitutive of the resulting algebra. Block scale, temporal depth, quotient relation, context selection, and admissible composition jointly determine which algebra becomes available. The natural object of study is therefore not simply $\mathcal{A}(F)$, but

$$\mathcal{A}(F, \pi),$$

where F is the substrate and π is the observation policy.

Third, algebraic structure forms a spectrum across scale rather than a single fixed theory. For a given cellular automaton,

$$\mathfrak{A}(F) = (\mathcal{A}_1(F), \mathcal{A}_2(F), \mathcal{A}_3(F), \dots).$$

These algebras need not grow monotonically. Laws and symmetries may be born, preserved, transformed, or extinguished as the scale changes.

Fourth, several distinct forms of algebraic emergence can be separated. Rule 178 provides a candidate for strong Algebraic Abiogenesis because a semilattice operation appears in an exact block quotient while the relevant microscopic binary minors lack commutativity. Rule 30 exhibits transformation-space emergence because its two-cell contexts generate the full transformation monoid on four states. Rules 60 and 90 exhibit algebraic morphogenesis, in which elementary additive structure reorganizes into scale-dependent affine and linear symmetry groups.

Fifth, arithmetic properties of scale can control emergent algebra. In additive cellular automata, Frobenius decomposition implies that multiplication of block size by the field characteristic preserves a nontranslation symmetry core while enlarging the translation module. For Rule 60, the odd part of the block size determines the dimension of the emergent general linear group. Scale is therefore not merely a measure of resolution; it can be an algebraically causal variable.

Together, these claims support a revised research program. Algebraic Abiogenesis is not the search for mathematics arising from literal nothingness. It is the exact study of how new mathematical objects, signatures, transformations, and theories become organized from lower-level dynamics under certified policies of observation and composition.

2. Algebraic Abiogenesis: Definitions and Criteria

2.1 Algebraic objects, operations, transformations, and laws

Algebraic Abiogenesis concerns the formation of four distinct but related kinds of mathematical structure: objects, operations, transformations, and laws. These should not be conflated, because each can emerge at a different stage and under a different observational policy.

An algebraic object is a stable unit of mathematical treatment. It may be a microscopic state, a block of cells, an equivalence class of blocks, a persistent defect, a transformation, or an entire process. Object formation therefore precedes most ordinary algebra. Before one can ask

whether an operation is associative, one must already know what counts as an operand and when two apparent operands count as the same object.

An operation is a rule that combines or transforms such objects. In familiar algebra, the signature is specified in advance. A group is supplied with a binary operation, a lattice with two binary operations, and a module with addition and scalar multiplication. In Algebraic Abiogenesis, the signature is not assumed to be known. The discovered structure may contain a ternary operation, a family of unary context maps, a partial composition law, a multivalued reaction, or a quotient-induced binary term.

A transformation is a map from an object space to itself or to another object space. Transformations become especially important in cellular automata because fixing part of a local or supercell environment produces maps on the remaining variable block. These maps can be composed even when the original cellular rule was not presented as an algebra of transformations. Closure under composition then produces a transformation monoid, and its invertible elements form a group.

A law is an exact relation satisfied by the resulting objects, operations, or transformations. Examples include commutativity,

$$x \star y = y \star x,$$

associativity,

$$(x \star y) \star z = x \star (y \star z),$$

idempotence,

$$x \star x = x,$$

or equality between two composite transformations,

$$f \circ g \circ h = p \circ q.$$

A law may emerge even when the relevant operation itself was not microscopically available. Conversely, an operation may emerge without satisfying any familiar equational theory. Algebraic Abiogenesis therefore includes both operation formation and law formation.

The most general output is an algebraic presentation

$$\mathcal{A} = \langle O, \Sigma \mid R \rangle,$$

where O is the formed object system, Σ is the discovered signature of admissible operations or transformations, and R is the certified set of relations.

2.2 The substrate-policy-algebra relation

The central formal relation of the program is

$$\boxed{(F, \pi) \longmapsto \mathcal{A}(F, \pi)}$$

where F denotes the lower-level dynamical substrate and π denotes an observation or admissibility policy.

The substrate F specifies what the system does microscopically. In a cellular automaton, it includes the alphabet, lattice, neighborhood, and local update rule. The policy π specifies how higher-level mathematical structure is extracted. It may include:

$$\pi = (k, \tau, \sim, C, Q),$$

where k is a block scale, τ is an evolution time, $\sim$ is an equivalence relation, C is a class of admissible contexts, and Q is a quotient or readout rule.

Two different policies applied to the same cellular automaton can produce nonisomorphic algebras:

$$\mathcal{A}(F, \pi) \ncong \mathcal{A}(F, \rho).$$

This does not imply arbitrariness. The policies must be exact, explicit, reproducible, and compatible with the dynamics. A quotient policy must define a genuine congruence if it is used to form an exact quotient algebra. A context policy must identify which variables are fixed and which are permitted to vary. A temporal policy must specify how long the system evolves before the macrostate is read.

The observation policy is therefore not an informal viewpoint. It is part of the mathematical construction. Once F and π are fixed, the resulting algebra is objectively determined.

This formulation also clarifies the role of the observer. The observer does not invent whether a particular quotient is exact or whether a transformation group has a certain order. Those are consequences of the substrate and policy. The observer selects the questions and admissible distinctions, but the dynamics determines whether the proposed structure is mathematically valid.

The program thus rejects both naive realism and unrestricted constructivism. The higher algebra is not simply lying on the surface of the microscopic rule, but neither is it freely imposed. It is formed through an exact interaction between substrate and policy.

2.3 Strong Algebraic Abiogenesis

Strong Algebraic Abiogenesis occurs when a macro-level algebraic operation or law becomes exactly available even though it is absent from the relevant microscopic comparison structure.

Let $\mathcal{M}(F)$denote the microscopic algebra induced directly by the local rule, and let $\mathcal{A}(F,\pi)$denote an algebra obtained through a nontrivial policy. A strong emergence event occurs when there exists a law Lsuch that

$$\mathcal{A}(F,\pi) \vDash L,$$

but

$$\mathcal{M}(F) \nvDash L$$

under the appropriate microscopic term interpretation.

This requires a carefully chosen comparison class. It is not enough to observe that the microscopic rule was not named “semilattice.” One must show that the relevant microscopic term operations do not already satisfy the law.

For example, suppose the macroalgebra contains a binary operation $\wedge$satisfying commutativity, associativity, and idempotence. To claim strong Algebraic Abiogenesis, one should examine the microscopic binary minors and short microscopic term operations. If every microscopic binary minor is noncommutative or merely a projection, then the macro-level semilattice is not a trivial relabeling of an existing microscopic operation.

Strong Algebraic Abiogenesis therefore requires a noninheritance certificate. The certificate identifies the microscopic operation class against which novelty is tested and proves that the macro-law is absent there.

The strongest form can be written as

$$L \in \mathrm{Th}(\mathcal{A}(F,\pi)) \setminus \mathrm{Th}(\mathcal{M}(F)),$$

where $\mathrm{Th}(\mathcal{A})$denotes the equational theory of $\mathcal{A}$.

This is stronger than ordinary coarse-graining. A coarse-grained variable may simply summarize microscopic behavior. Strong Algebraic Abiogenesis requires the appearance of a genuinely new exact algebraic organization relative to the chosen microscopic language.

2.4 Weak Algebraic Abiogenesis and algebraic morphogenesis

Weak Algebraic Abiogenesis occurs when the microscopic substrate already possesses relevant algebraic structure, but scale, context, or composition reorganizes it into a substantially richer higher-level form.

Linear cellular automata provide the clearest examples. Their local rules are built from addition over a finite field. The microscopic substrate therefore already contains algebra. When block contexts generate affine groups, general linear groups, or semidirect products, the resulting structures cannot be described as arising from an algebra-free base.

Nevertheless, the higher groups are not explicitly present as microscopic objects. A local XOR rule does not itself list the elements of

$$AGL(k, 2)$$

or specify the action

$$\mathbb{F}_2^k \rtimes GL(m, 2).$$

These structures appear only after blocks are formed, contexts are composed, and the invertible transformation closure is computed.

This process is better called algebraic morphogenesis:

$$\text{elementary algebra} \longrightarrow \text{higher organized algebra}.$$

The distinction from strong Algebraic Abiogenesis is one of inheritance. In weak abiogenesis, the macrostructure is constructed from recognizable microscopic algebraic ingredients. In strong abiogenesis, the macro-law is absent from the selected microscopic term structure.

Weak abiogenesis is not scientifically inferior. Many important forms of emergence involve reorganization rather than creation of entirely new primitives. Biological development, crystallization, and collective synchronization all transform existing local mechanisms into higher order. Similarly, additive cellular automata can reorganize simple field addition into large, scale-sensitive symmetry groups.

The category of weak abiogenesis is therefore necessary to avoid the false choice between absolute novelty and mere restatement.

2.5 Algebraic revelation and inherited structure

Algebraic revelation is the weakest category. It occurs when the discovery procedure merely exposes a structure already explicitly encoded in the substrate.

For example, if a binary operation table is supplied and the program reports the same table under renamed symbols, no new algebra has formed. If a linear cellular automaton is represented directly by matrices and the program announces matrix multiplication, the result may be useful but is not abiogenic. Likewise, recovering parity from an affine rule may amount to identifying an invariant already transparent from its defining equation.

The distinction can be represented as follows:

Strong abiogenesis:	new exact macro-law absent microscopically,
Weak abiogenesis:	new higher organization of inherited algebra,
Revelation:	explicit microscopic algebra rediscovered.

These categories may depend on the representation chosen for the substrate. A rule given only as an anonymous lookup table may hide linearity from the program even though a human later recognizes it. Computationally, the system may have discovered an affine structure without being told to search for one. Mathematically, however, the structure was inherited from the rule.

The research program must distinguish computational surprise from structural novelty. A machine may find something unexpected to its designers without discovering a genuinely new algebraic level.

Algebraic revelation remains useful as calibration. A discovery system should be able to recover known algebraic properties from rules where the answer is understood. But calibration results should not be presented as evidence of strong mathematical emergence.

2.6 Exactness, noninheritance, threshold birth, and certification

A convincing abiogenic result should satisfy four principal criteria: exactness, noninheritance, threshold birth, and certification.

Exactness means that the emergent structure holds for every admissible microscopic representative, not merely with high probability or in sampled simulations. If a partition defines a quotient algebra, then the output class must depend only on the input classes:

$$x_i \sim y_i \text{ for all } i \Longrightarrow F(x_1, \dots, x_n) \sim F(y_1, \dots, y_n).$$

A statistical regularity may be scientifically interesting, but it does not yet define an exact algebra.

Noninheritance means that the law is absent from a clearly stated microscopic comparison class. This may involve checking all binary minors, all terms up to a bounded depth, or a theoretically complete family of microscopic operations. The stronger the comparison, the stronger the claim.

Threshold birth means that the structure first appears at a nontrivial scale or policy. If a law holds microscopically, its appearance at a larger scale is not a birth event. A meaningful threshold has the form

$$\mathcal{A}_k \nvDash L \text{for } k < k_0,$$

but

$$\mathcal{A}_{k_0} \vDash L.$$

Persistence beyond the threshold further strengthens the result:

$$\mathcal{A}_k \vDash L \text{for all } k \geq k_0.$$

Certification means that the result is accompanied by independently checkable evidence. Depending on the result, the certificate may include:

- a complete operation table;
- a congruence partition;
- a list of generating transformations;
- a closure computation;
- a Smith normal form;
- an exact group-order calculation;
- a symbolic proof;
- or exhaustive enumeration over a finite search space.

The ideal workflow combines computation and proof. Computation identifies a candidate structure. Exact certificates verify the finite claim. Mathematical analysis then explains why the pattern persists beyond the computed range.

A strong abiogenic claim therefore has the form

$$\boxed{\text{exact} + \text{noninherited} + \text{thresholded} + \text{certified.}}$$

2.7 Algebraic genesis, persistence, and extinction

Once algebras are indexed by policy or scale, their laws can be studied dynamically.

Let

$$\mathcal{A}_k(F)$$

denote the algebra extracted at scale k. For a law L, define its support set

$$S_L(F) = \{k : \mathcal{A}_k(F) \vDash L\}.$$

The structure of $S_L(F)$describes the life history of the law across scale.

Genesis occurs when a law first becomes valid:

$$k_0 = \min S_L(F).$$

Then

$$\mathcal{A}_{k_0}(F) \vDash L,$$

while the law fails at all smaller admissible scales.

Persistence occurs when a law continues to hold across later scales or along a specified arithmetic family of scales. Persistence may be consecutive,

$$k_0, k_0 + 1, k_0 + 2, \ldots,$$

or arithmetic,

$$m, pm, p^2m, \ldots.$$

Extinction occurs when a law or symmetry present at one scale disappears at a later scale:

$$\mathcal{A}_k(F) \vDash L, \mathcal{A}_{k+1}(F) \nvDash L.$$

Rebirth is also possible:

$$L \in \mathrm{Th}(\mathcal{A}_k), L \notin \mathrm{Th}(\mathcal{A}_{k+1}), L \in \mathrm{Th}(\mathcal{A}_{k+2}).$$

These phenomena show that algebraic complexity is not necessarily monotone under coarse observation. Larger blocks can reveal new symmetries, destroy earlier symmetries, or reorganize them into different group types.

The resulting object is not a single algebra but a theory-valued sequence:

$$\mathfrak{A}(F) = (\mathcal{A}_1(F), \mathcal{A}_2(F), \mathcal{A}_3(F), \dots).$$

One may then study transition maps, quotient relations, embeddings, and recurring patterns between scales.

The language of genesis and extinction should not be interpreted biologically in a literal sense. It is a precise statement about validity under changing policies. A law is born when it first becomes exact in the policy-indexed family. It dies when it ceases to hold. This language turns algebraic emergence into a subject with its own dynamics, thresholds, phase transitions, and recurrence structures.

3. Cellular Automata as a Laboratory for Mathematical Emergence

3.1 Elementary cellular automata and anonymous local rules

Elementary cellular automata provide one of the smallest complete laboratories in which higher mathematical organization can be studied without ambiguity about the underlying dynamics. Each system consists of a one-dimensional lattice of binary cells, a radius-one neighborhood, and a deterministic local rule. The state of cell $i$at time $t+1$is determined by the triple formed by cells $i-1$, i, and $i+1$at time t:

$$x_i(t+1) = f(x_{i-1}(t), x_i(t), x_{i+1}(t)),$$

with

$$f: \{0,1\}^3 \to \{0,1\}.$$

Because there are eight possible binary neighborhoods and two possible outputs for each, there are

$$2^8 = 256$$

elementary rules.

The conventional numbering of these rules assigns a decimal integer to each eight-bit lookup table. That numbering is useful for communication but irrelevant to the abiogenic method. The discovery system need not know that a particular lookup table is called Rule 30, Rule 60, Rule 90, Rule 110, or Rule 178. It can receive the eight outputs as an anonymous deterministic mapping.

This anonymity is methodologically important. If the system is told that a rule is additive, chaotic, universal, affine, or particle-rich, then part of the desired mathematical interpretation has already been supplied. In the strongest experiments, the rule is treated simply as a black-box local operation whose global consequences are computed exactly.

The local rule is nevertheless not structureless. It presupposes binary states, an ordered neighborhood, discrete time, and deterministic evolution. Cellular automata are therefore not nonmathematical substrates. Their value lies in the gap between the simplicity of their microscopic description and the diversity of exact higher structures that can arise from it.

A single local lookup table can support periodic domains, mobile defects, exact block transformations, quotient operations, context-generated monoids, symmetry groups, and arithmetic scale spectra. The microscopic rule specifies none of these in its native vocabulary. They become available only after additional structures of observation and composition are introduced.

Elementary cellular automata are therefore sufficiently simple for exhaustive study and sufficiently rich for genuine algebraic organization to appear.

3.2 Microscopic states, neighborhoods, and spacetime evolution

A global configuration is a bi-infinite binary sequence

$$x(t) \in \{0,1\}^{\mathbb{Z}}.$$

The local rule induces a global map

$$F: \{0,1\}^{\mathbb{Z}} \to \{0,1\}^{\mathbb{Z}}.$$

Repeated application produces the orbit

$$x(0), F(x(0)), F^2(x(0)), \ldots.$$

Represented graphically, these configurations form a spacetime diagram in which horizontal position denotes space and vertical position denotes time. Such diagrams often display structures that appear to move, collide, replicate, or form regular backgrounds.

Visual appearance alone, however, is not sufficient for Algebraic Abiogenesis. A pattern may look like an object without supporting an exact equivalence class. Two apparent particles may resemble one another while differing in phase or future behavior. A collision may appear deterministic under one coarse observation but split into several outcomes when hidden distinctions are retained.

The basic microscopic evolution is exact. Therefore any higher object or law should ultimately be grounded in exact properties of spacetime evolution.

Finite propagation speed is particularly important. In a radius-one cellular automaton, information can travel at most one cell per time step. The state of a cell after $t$steps depends only on an interval of radius $t$in the initial configuration. This causal structure allows finite blocks to be evolved independently when sufficient boundary data are supplied.

For a finite interval I, its future over time $t$depends on the enlarged interval

$$I + [-t, t].$$

This makes exact supercell constructions possible. It also provides a clear distinction between internal and contextual information. The central block is the candidate object; the surrounding blocks supply the environment through which it is transformed.

The spacetime view supports several different forms of object formation. A persistent moving defect can be defined through recurrence up to translation. A block can be treated as a finite macrostate. An equivalence class of blocks can become a quotient object. A spacetime slab can become a morphism between incoming and outgoing boundary states.

The cellular automaton therefore does not dictate a single ontology. It supports several exact ontologies depending on which spacetime structures are selected for mathematical treatment.

3.3 Exact block dynamics and supercell operations

The central construction of the present framework is the exact block or supercell operation.

Fix a block size k. Let

$$B_k = \{0,1\}^k$$

be the set of all binary blocks of length k. Place three such blocks side by side:

$$u\,v\,w, u, v, w \in B_k.$$

Evolve the resulting $3k$-cell configuration for $k$time steps under the cellular automaton. By finite propagation speed, the central $k$output cells depend only on those three initial blocks. Reading the central block defines a ternary operation

$$\Phi_{F,k}: B_k^3 \to B_k.$$

This operation is exact. It is not estimated from samples, inferred statistically, or chosen to approximate long-term behavior. Every possible triple of input blocks can be enumerated, and the output is determined uniquely by the microscopic rule.

The scale-k block algebra is

$$\mathcal{B}_k(F) = (B_k, \Phi_{F,k}).$$

The microscopic cellular automaton corresponds to the scale-one case:

$$\mathcal{B}_1(F) = (\{0,1\}, f).$$

At larger scales, the state set grows exponentially:

$$| B_k | = 2^k.$$

The operation table grows even faster, since it contains

$$| B_k |^3 = 2^{3k}$$

entries. Nevertheless, low block sizes remain computationally accessible and can reveal exact structures absent at scale one.

The use of $k$time steps for blocks of width $k$is natural because it produces a geometrically self-similar supercell construction. Other temporal depths can also be studied. More generally, one may define

$$\Phi_{F,k,\tau}: B_k^m \to B_k$$

for chosen block width, elapsed time, and number of contextual blocks. The present work concentrates on the simplest exact construction that preserves the local causal geometry.

Once the block operation has been formed, it can be studied as a finite algebra independently of how it was generated. Its congruences can be enumerated. Its term operations can be constructed. Its unary translations can be composed. Its automorphism group can be computed. Its quotient structures can be compared across scales.

This is the principal transition from dynamics to algebra:

$$F \longrightarrow \Phi_{F,k} \longrightarrow \mathcal{B}_k(F).$$

The local rule becomes a family of finite algebras indexed by scale.

3.4 Observation scale as part of the mathematical experiment

Block size is not merely a numerical convenience. It determines which configurations count as single objects and which transformations become elementary.

At scale one, the observer distinguishes only the two cell states. At scale two, the object space contains four blocks:

$$00,01,10,11.$$

At scale three, there are eight objects. Each increase in scale changes the state space, the induced operation, the available congruences, and the group of context-generated transformations.

The observation scale is therefore algebraically causal. Changing $k$can create operations that do not exist at smaller scales, destroy symmetries previously present, or reorganize the transformation structure into a different group.

This differs from the common expectation that larger scales merely simplify or average microscopic behavior. Exact block formation can increase algebraic richness. Rule 30, for example, has only a small microscopic context group but generates the full symmetric group on

four block states at scale two. Conversely, some rules possess non-Abelian symmetry at scale two and lose it at scale three.

The appropriate question is not whether the larger scale is more complex in an absolute sense. It is whether the new scale supports a different exact algebraic organization.

Scale can also have arithmetic effects. In additive cellular automata over characteristic two, block sizes with different odd parts can generate different linear symmetry cores. Doubling the scale may preserve one part of the group while enlarging another. Odd scales and powers of two may therefore belong to different algebraic regimes.

The scale variable should consequently be incorporated into the formal policy:

$$\pi_k = (k, \tau_k, \sim_k, C_k).$$

An emergent law is always a statement about a particular scale-indexed algebra:

$$\mathcal{A}(F, \pi_k) \vDash L.$$

Without specifying k, claims about the algebra of a cellular automaton may be incomplete or misleading.

3.5 Context selection, quotienting, and admissibility policies

Block formation alone does not determine the final emergent algebra. Additional policies decide which distinctions and transformations are retained.

A context policy fixes some arguments of the block operation while allowing another to vary. For the ternary operation

$$\Phi_{F,k}(u, v, w),$$

one may fix $u$and $w$and vary v, producing a unary map

$$\lambda_{u,w}(v) = \Phi_{F,k}(u, v, w).$$

Alternatively, one may vary the left or right argument while fixing the other two. The collection of all such maps generates a transformation monoid under composition.

A quotient policy identifies block states according to an equivalence relation

$$\theta \subseteq B_k \times B_k.$$

For the quotient to support an exact induced operation, θmust be a congruence:

$$u_i \equiv_\theta v_i \text{for each input coordinate}$$

must imply

$$\Phi_{F,k}(u_1, u_2, u_3) \equiv_\theta \Phi_{F,k}(v_1, v_2, v_3).$$

Only then is the quotient operation well defined.

The Rule 178 construction uses a policy that distinguishes three block classes:

$$0^k, 1^k, \text{mixed blocks.}$$

The exactness of the quotient is not assumed from this intuitive classification. It must be verified that every triple of representatives from the same classes produces outputs in the same class.

An admissibility policy may also specify whether reflection, translation, temporal phase, or internal permutations are ignored. In particle-based approaches, two localized patterns may be identified up to translation. In context-monoid approaches, the exact labels of block states may be retained because the generated transformations act on them differently.

The general policy-indexed construction is therefore

$$\mathcal{A}(F, \pi),$$

where πgoverns object formation, equivalence, context selection, and composition.

A policy is admissible when it produces an exact, reproducible structure. It need not be unique. Several admissible policies may coexist, generating different algebras from the same substrate.

3.6 Why a cellular automaton supports an ecology rather than one algebra

The cellular automaton should not be regarded as possessing one privileged emergent algebra.

At one scale and under one quotient, it may support a semilattice. Under a context policy, it may generate a transformation monoid. Its invertible transformations may form a permutation group. A particle policy may produce a reaction system and a conservation group. A congruence analysis may produce a lattice of observational resolutions.

The natural mathematical object is therefore an ecology:

$$\mathfrak{E}(F) = \{\mathcal{A}(F, \pi) : \pi \in \Pi_F\},$$

where $\Pi_F$is the family of admissible policies for the substrate F.

This ecology contains structures of different kinds. It is not simply a list of groups or rings. It may include

finite quotient algebras,
semilattices and other term reducts,
transformation monoids,
permutation groups,
congruence lattices,
reaction presentations,
process categories,
scale-indexed spectra.

Relations among these structures are themselves mathematically significant. A quotient may reduce a transformation monoid. A context monoid may act on a congruence lattice. A scale-doubling map may preserve a symmetry core while enlarging a translation module. A law present in one quotient may fail in another.

The ecology perspective resolves a common ambiguity in discussions of emergence. Asking "What algebra does Rule 30 have?" is too coarse. More precise questions are:

Which algebra is extracted at scale k?
Under which quotient policy?
Using which admissible contexts?
With which notion of equivalence?
Which laws survive changes of policy?

A law that recurs across many unrelated policies may be considered robust. A law that appears only under one fragile quotient may be policy-specific. A group structure that persists along an infinite arithmetic family of scales may indicate a deeper organizing principle.

Cellular automata therefore provide more than examples of emergent algebra. They provide a finite experimental setting in which entire mathematical ecologies can be generated, compared, certified, and classified.

4. Exact Block Algebras and Microscopic Noninheritance

4.1 Construction of the scale-k block algebra

Let F be an elementary cellular automaton with local rule

$$f: \{0,1\}^3 \to \{0,1\}.$$

For each block length $k \geq 1$, define the macrostate space

$$B_k = \{0,1\}^k.$$

An element of B_k is a finite binary word

$$u = (u_0, u_1, \dots, u_{k-1})$$

that is treated as one scale-k object.

To determine how such objects interact, place three blocks $u, v, w \in B_k$ consecutively in space:

$$u\, v\, w.$$

Evolve the resulting $3k$-cell configuration for k cellular-automaton steps. Because the automaton has radius one, the central k output cells depend only on the original three blocks. The causal cone of the central block remains within their combined support.

Reading the central output block defines

$$\Phi_{F,k}(u, v, w) \in B_k.$$

Thus

$$\Phi_{F,k}: B_k^3 \to B_k$$

is a well-defined ternary operation.

The associated finite algebra is

$$\mathcal{B}_k(F) = (B_k, \Phi_{F,k}).$$

At scale one,

$$\mathcal{B}_1(F) = (\{0,1\}, f),$$

so the microscopic local rule is the first member of the block-algebra family.

The complete scale-indexed family is

$$\mathfrak{B}(F) = (\mathcal{B}_1(F), \mathcal{B}_2(F), \mathcal{B}_3(F), \dots).$$

This family is not a sequence of approximations. Each algebra is exact for its chosen block scale and readout policy. The larger algebras contain more states and more complicated operations, but every table entry is determined by finite microscopic evolution.

The operation table of $\mathcal{B}_k(F)$contains

$$| B_k |^3 = 2^{3k}$$

entries. For $k = 3$, there are eight block states and

$$8^3 = 512$$

input triples. This scale is large enough to support nontrivial quotient structures while remaining small enough for complete enumeration over all elementary cellular automata.

4.2 Central-block evolution as a ternary operation

The ternary character of $\Phi_{F,k}$reflects the causal geometry of the one-dimensional radius-one automaton. The future of the central block depends on the block itself and on one neighboring block on each side.

The three arguments may be interpreted as

$$\Phi_{F,k}(\text{left context,central object,right context}).$$

This interpretation is useful but should not obscure the algebraic fact that all three arguments belong to the same set B_k. The resulting structure is a one-sorted ternary algebra.

The operation need not satisfy any familiar identity. It may be asymmetric in its arguments:

$$\Phi_{F,k}(u, v, w) \neq \Phi_{F,k}(w, v, u).$$

It may fail every standard associativity analogue. It may possess projections, absorbing elements, constants, or unexpected term operations.

The ternary signature is not imposed because ternary algebra is the desired target. It arises directly from the geometry of the construction. This is methodologically important. Algebraic Abiogenesis should preserve the native arity supplied by the substrate before searching for derived operations of lower or higher arity.

Binary operations may later be formed by identifying or fixing arguments. Examples include

$$\mu_1(x, y) = \Phi_{F,k}(x, x, y),$$
$$\mu_2(x, y) = \Phi_{F,k}(x, y, x),$$

and

$$\mu_3(x, y) = \Phi_{F,k}(y, x, x).$$

One may also fix a block $c \in B_k$and define

$$\mu_c(x, y) = \Phi_{F,k}(x, c, y).$$

These operations are derived from the block algebra rather than supplied independently.

The ternary operation therefore acts as an algebraic generator. Its term clone contains operations of many arities formed through substitution and composition. The search for emergent laws can be conducted within this generated term structure or within quotients of it.

4.3 Congruences and exact quotient formation

An equivalence relation θon $B_k$partitions the block states into observational classes. The relation can support an exact quotient algebra only if it is compatible with $\Phi_{F,k}$.

The compatibility condition is

$$u_1 \equiv_\theta v_1, u_2 \equiv_\theta v_2, u_3 \equiv_\theta v_3$$

implies

$$\Phi_{F,k}(u_1, u_2, u_3) \equiv_\theta \Phi_{F,k}(v_1, v_2, v_3).$$

An equivalence relation satisfying this condition is a congruence of the block algebra.

If $\theta$is a congruence, the quotient set

$$B_k/\theta$$

inherits an exact ternary operation

$$\bar{\Phi}_{F,k}([u]_\theta, [v]_\theta, [w]_\theta) = [\Phi_{F,k}(u, v, w)]_\theta.$$

The compatibility condition guarantees that the output class does not depend on the chosen representatives.

The quotient algebra is

$$\mathcal{B}_k(F)/\theta = (B_k/\theta, \bar{\Phi}_{F,k}).$$

This is stronger than approximate coarse-graining. In an approximate macrodescription, members of one class may occasionally produce outputs in different classes. In an exact quotient, every representative behaves identically at the quotient level for every admissible context.

The set of all congruences of a finite algebra forms a lattice under intersection and generated join:

$$\mathrm{Con}\,\mathcal{B}_k(F).$$

The equality relation is the least congruence, and the universal relation is the greatest. Intermediate congruences represent nontrivial exact observation policies.

A complete congruence search asks which partitions of $B_k$satisfy compatibility. For eight states at scale three, the total number of set partitions is

$$B_8 = 4140,$$

where $B_8$is the eighth Bell number. This is small enough for exhaustive testing against all 512 operation-table entries.

The output is not one hand-selected quotient but the complete congruence spectrum of the block algebra.

4.4 Binary minors and term-derived operations

The quotient operation remains ternary, but familiar algebraic structures may appear in its binary minors.

Given a ternary operation m, its six elementary binary minors are obtained by substituting $x$and $y$into the three argument positions while requiring both variables to occur:

$$m(x,x,y), \quad m(x,y,x), m(y,x,x),$$
$$m(x,y,y), \quad m(y,x,y), m(y,y,x).$$

Each minor defines a binary operation on the same state set.

A quotient ternary operation may therefore contain a semilattice, quasigroup, band, rectangular band, projection algebra, or another binary structure as a term reduct.

For a binary operation $\star$, the principal identities tested include

$$x \star y = y \star x,$$
$$(x \star y) \star z = x \star (y \star z),$$
$$x \star x = x,$$

and possible identity or absorbing-element laws.

These tests are exact because the state set is finite. Commutativity requires checking all ordered pairs. Associativity requires checking all ordered triples. No sampling is necessary.

The search should not begin by assuming that a semilattice is the desired result. It should construct all valid quotients, derive all elementary binary minors, and classify the identities that occur. A semilattice is recognized only after its operation table has been generated and tested.

More complex term operations can be generated recursively. If $t_1, t_2, t_3$are terms, then

$$m(t_1, t_2, t_3)$$

is another term. Enumeration by depth can reveal laws that do not occur among the elementary minors.

However, the simplest minors are especially valuable because they provide transparent certificates. A three-state binary operation can be displayed in a 3×3table and independently checked by inspection.

4.5 Testing whether a macro-law already exists microscopically

The appearance of a law at block scale does not by itself establish strong Algebraic Abiogenesis. The same law may already occur in the microscopic algebra.

The microscopic operation is

$$f: \{0,1\}^3 \to \{0,1\}.$$

Its binary minors are

$$f(x,x,y), f(x,y,x), f(y,x,x), f(x,y,y), f(y,x,y), f(y,y,x).$$

These can be classified exhaustively.

Suppose a quotient at scale three contains a commutative, associative, idempotent binary minor. To determine whether the law is inherited, one must ask whether any corresponding microscopic minor also has those properties.

If none is commutative, then no elementary microscopic minor can be a semilattice. This yields a first-level noninheritance certificate.

A stronger comparison may examine the entire microscopic term clone rather than only the six elementary minors. That asks whether any binary operation generated by arbitrary composition of f satisfies the macro-law.

For a two-element microscopic set, the total number of binary operations is only

$$2^{2^2} = 16.$$

The generated binary term clone can therefore be computed completely. If no commutative idempotent associative operation occurs there, the noninheritance claim becomes stronger.

The comparison must preserve the intended relationship between micro and macro terms. A macro-operation formed by a particular variable identification should ideally be compared with the analogous microscopic identification. A broader clone comparison may then be added as a robustness test.

Microscopic noninheritance does not imply logical independence from the microscopic rule. The macro-law remains a consequence of the rule combined with the block and quotient policy. The claim is narrower:

the macro-law is not available as an operation in the selected microscopic algebraic language.

This is the correct sense in which a law can be born without violating reduction.

4.6 Complete congruence searches over elementary cellular automata

The finite size of scale-three block algebras permits a complete search across the entire elementary cellular-automaton rule space.

For each rule F:

1. Construct the scale-three block set
$$B_3 = \{0,1\}^3.$$
2. Compute the full ternary table
$$\Phi_{F,3}: B_3^3 \to B_3.$$
3. Enumerate all 4140 partitions of the eight-state set.
4. Test every partition for congruence compatibility.
5. Construct each nontrivial quotient operation.
6. Derive its six elementary binary minors.
7. Test each minor for specified algebraic laws.
8. Compare successful macro-laws with the corresponding microscopic minors.
9. Canonicalize isomorphic results under relabeling of quotient classes.
10. Store exact certificates for every surviving witness.

This process transforms the search from anecdotal experimentation into finite classification.

The output includes several levels of information:

number of congruences per rule,
quotient cardinalities,
quotient-operation isomorphism types,
binary-minor law profiles,
microscopic inheritance status,
scale thresholds,
symmetry-class relationships.

A complete search also produces negative results. If no quotient of a particular size supports a quandle, heap, discriminator, or semilattice law under the tested construction, that absence constrains the space of plausible emergent structures.

Exhaustiveness matters because isolated positive examples can be misleading. A structure that initially appears exceptional may occur in dozens of equivalent rules or arise from a trivial quotient. Complete enumeration reveals whether a witness is unique, generic, symmetry-related, or degenerate.

The Rule 178 semilattice result is important partly because it survived this broader comparison. It was not selected from one hand-designed partition after examining one rule. It emerged as the unique witness under the specified complete three-state, scale-three law-genesis search.

4.7 Certificates of genuine algebraic law birth

A convincing claim of algebraic law birth should be accompanied by a certificate containing enough information for independent reconstruction.

For a quotient-based witness, the certificate should include:

the microscopic rule table,
the block size and evolution time,
the complete quotient partition,
the induced quotient operation table,
the derived binary operation table,
the verified identities,
the microscopic comparison operations,
the noninheritance result.

For threshold birth, the certificate should additionally show failure below the threshold and exact success at the threshold:

$$\mathcal{B}_{k_0-1}(F) \not\models L, \mathcal{B}_{k_0}(F)/\theta \models L.$$

If persistence is claimed, computed evidence at larger scales should be supplemented by a proof whenever possible.

The ideal result has three layers.

The first is computational discovery. Exhaustive enumeration finds a quotient and operation satisfying an unexpected law.

The second is finite certification. Complete tables, partitions, and identity checks establish that the specific instance is correct.

The third is structural explanation. A theorem identifies the dynamical mechanism responsible for the law and proves persistence beyond the enumerated range.

This three-layer standard distinguishes a genuine mathematical contribution from pattern recognition:

$$\boxed{\text{discovery} \longrightarrow \text{certificate} \longrightarrow \text{theorem.}}$$

The purpose of the exact block-algebra method is therefore not merely to produce interesting tables. It is to create a reproducible bridge from anonymous local dynamics to formally certified algebraic organization.

5. Rule 178 and the Threshold Birth of a Semilattice

5.1 The microscopic asymmetry of Rule 178

Rule 178 is an elementary binary cellular automaton whose local update operation

$$f: \{0,1\}^3 \to \{0,1\}$$

is defined by the output string

$$10110010$$

under the conventional neighborhood ordering

$$111,110,101,100,011,010,001,000.$$

Its local rule can be written compactly as

$$f(a,b,c) = \begin{cases} a, & a = c, \\ 1-b, & a \neq c. \end{cases}$$

This expression shows that the rule distinguishes the outer cells. When the left and right neighbors agree, their common value is copied. When they disagree, the central cell is complemented.

The microscopic operation is not commutative in any natural binary sense. The six elementary binary minors obtained by identifying two of the three arguments reduce to left or right projections. For example,

$$f(x, x, y) = y,$$

while another substitution yields

$$f(x, y, y) = x.$$

Across the six possible placements of $x$and y, three minors return the left variable and three return the right variable. None is commutative unless the input values happen to agree.

A projection operation such as

$$p(x, y) = x$$

is associative and idempotent, but it is not commutative:

$$p(x, y) = x, p(y, x) = y.$$

Thus no microscopic binary minor forms a semilattice.

The asymmetry of the local rule makes Rule 178 a particularly useful test case. If a commutative operation later appears at block scale, it cannot be dismissed as the straightforward preservation of microscopic commutativity. The higher operation must arise from the combined effects of spatial aggregation, temporal evolution, and quotient identification.

5.2 The homogeneous-mixed-homogeneous quotient policy

For a block length k, let

$$B_k = \{0,1\}^k.$$

Define an observation policy

$$P_k: B_k \to \{0,1,2\}$$

by

$$P_k(u) = \begin{cases} 0, & u = 0^k, \\ 2, & u = 1^k, \\ 1, & u \text{ contains both } 0 \text{ and } 1. \end{cases}$$

The three quotient classes are therefore:

$$[0] = \{0^k\},$$
$$[2] = \{1^k\},$$

and

$$[1] = B_k \setminus \{0^k, 1^k\}.$$

The class labels are conventional. Class 0represents a homogeneous zero block, class 2represents a homogeneous one block, and class 1contains every mixed block.

This policy discards almost all microscopic information. At scale k, there are 2^kbinary blocks, but only three observed macroclasses. The mixed class alone contains

$$2^k - 2$$

different microstates.

Such compression does not automatically define a quotient algebra. For the policy to be exact, every triple of blocks with the same input class labels must produce central output blocks with the same output class label after $k$steps.

Let

$$\Phi_{178,k}: B_k^3 \to B_k$$

be the exact scale-k supercell operation. The required condition is that there exist a ternary operation

$$m: \{0,1,2\}^3 \to \{0,1,2\}$$

such that

$$P_k\left(\Phi_{178,k}(u,v,w)\right) = m\left(P_k(u), P_k(v), P_k(w)\right)$$

for every

$$u, v, w \in B_k.$$

When this condition holds, the homogeneous-mixed-homogeneous partition is a congruence of the block algebra.

The discovered quotient operation is

$$m(a,b,c) = \begin{cases} 0, & a = b = c = 0, \\ 2, & a = b = c = 2, \\ 1, & \text{otherwise.} \end{cases}$$

Thus the central output is homogeneous zero only when all three input blocks are homogeneous zero. It is homogeneous one only when all three are homogeneous one. Every other combination produces a mixed central block.

5.3 Failure at block sizes one and two

The quotient does not hold at every scale. Its failure at block sizes one and two is essential because it establishes a genuine threshold rather than the persistence of a microscopic classification.

At scale one, every block consists of a single bit. There are no mixed blocks, so the proposed three-class state space is not realized. Only the homogeneous classes $0$and 2exist. The intended three-element quotient cannot therefore be formed.

At scale two, mixed blocks exist:

$$01,10.$$

However, the homogeneous-mixed-homogeneous partition is not dynamically stable. Some triples containing mixed or conflicting block classes can evolve to a homogeneous central output after two steps.

Consequently, two triples with identical quotient labels can produce outputs belonging to different quotient classes. The compatibility condition fails:

$$P_2(u_i) = P_2(v_i) \text{for } i = 1,2,3$$

does not always imply

$$P_2\left(\Phi_{178,2}(u_1,u_2,u_3)\right) = P_2\left(\Phi_{178,2}(v_1,v_2,v_3)\right).$$

This failure is not numerical noise. It is an exact obstruction.

The scale-two block is too narrow to guarantee that a nonhomogeneous causal configuration leaves a visible internal domain wall after two steps. Opposing influences can meet in a way that removes the distinction detected by the quotient policy.

The threshold result can therefore be stated as

P_k is not an exact three-state quotient for $k < 3$.

The semilattice does not merely become easier to notice at scale three. Its supporting quotient first becomes mathematically well defined there.

5.4 Exact formation of the quotient at block size three

At scale three,

$$B_3 = \{0,1\}^3$$

contains eight blocks. The proposed partition is

$$\{000\},$$
$$\{001,010,011,100,101,110\},$$
$$\{111\}.$$

The complete ternary supercell table contains

$$8^3 = 512$$

input triples. Exhaustive evaluation shows that the output class depends only on the three input classes.

Specifically,

$$(000, 000, 000)$$

evolves to a homogeneous-zero central block, while

$$(111, 111, 111)$$

evolves to a homogeneous-one central block. Every other triple of block classes produces a mixed central block.

Therefore,

$$P_3\left(\Phi_{178,3}(u,v,w)\right) = m\left(P_3(u), P_3(v), P_3(w)\right)$$

holds for all 512 input triples.

The quotient is exact:

$$\mathcal{B}_3(178)/\theta = (\{0,1,2\}, m),$$

where $\theta$is the homogeneous-mixed-homogeneous congruence.

This is the first scale at which the three observed classes become closed under the induced dynamics.

The scale-three threshold is especially meaningful because the quotient was not hand-selected after inspecting one successful trajectory. The complete search considered every three-class partition of the eight block states and tested every possible input triple. The successful partition appeared within a rule-wide and partition-wide exhaustive search.

The result is therefore not based on a favored initial condition. It is a finite theorem about the entire scale-three block algebra.

5.5 The emergent commutative, associative, idempotent operation

The quotient operation $m$is ternary. A binary operation is obtained through the minor

$$x \wedge y = m(x, x, y).$$

Using the quotient rule for m, the operation satisfies

$$x \wedge y = \begin{cases} 0, & x = y = 0, \\ 2, & x = y = 2, \\ 1, & \text{otherwise.} \end{cases}$$

Its operation table is

$$\begin{array}{cccc} \wedge & 0 & 1 & 2 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 2 \end{array}$$

This operation is commutative because the table is symmetric:

$$x \wedge y = y \wedge x.$$

It is idempotent because every diagonal entry returns its argument:

$$x \wedge x = x.$$

It is associative:

$$(x \wedge y) \wedge z = x \wedge (y \wedge z).$$

Associativity can be verified directly over the

$$3^3 = 27$$

possible triples. It also follows from the order-theoretic interpretation of the table.

The operation therefore satisfies the defining identities of a semilattice:

$$\begin{array}{cc} x \wedge y & = y \wedge x, \\ (x \wedge y) \wedge z & = x \wedge (y \wedge z), \\ x \wedge x & = x. \end{array}$$

None of these identities was supplied as a target during the construction of the block algebra or congruence search. The operation was generated from the quotient table and classified afterward.

The semilattice thus represents the birth of a familiar algebraic signature from a ternary macro-operation whose microscopic source has no commutative binary minor.

5.6 Semilattice structure and its order-theoretic interpretation

Every meet-semilattice defines a partial order by

$$x \leq y \Leftrightarrow x \wedge y = x.$$

For the discovered operation,

$$1 \wedge 0 = 1, 1 \wedge 2 = 1,$$

so

$$1 \leq 0, 1 \leq 2.$$

The homogeneous states $0$and $2$are incomparable because

$$0 \wedge 2 = 1,$$

which is neither 0nor 2.

The order therefore has the form

$$\begin{array}{c} 02 \\ \diagup\!\diagdown \\ 1 \end{array}$$

with mixedness as the common lower state and the two homogeneous states as incomparable maxima.

The meet has a clear interpretation. Two macrodescriptions retain a homogeneous value only when both agree on the same homogeneous state. Any disagreement, or any participation of a mixed block, yields the mixed class.

Thus

$$0 \wedge 0 = 0,$$
$$2 \wedge 2 = 2,$$

but

$$0 \wedge 2 = 1.$$

The semilattice operation acts as a consensus-with-conflict map. Exact homogeneous agreement is preserved. Every incompatible or uncertain combination collapses to mixedness.

The mixed state is absorbing:

$$1 \wedge x = 1$$

for all x. Once unresolved internal variation is present in one operand, the meet cannot certify global homogeneity.

This interpretation connects the quotient to information loss. The homogeneous classes carry definite global statements:

all zeroorall one.

The mixed class represents the failure of either homogeneous assertion. The meet operation returns the strongest common homogeneous assertion supported by both operands; when none exists, it returns mixedness.

The algebra is therefore not merely a formal table. It expresses how exact macroscopic certainty behaves under the Rule 178 block dynamics.

5.7 Domain-wall dynamics and the Rule 222 representation

The persistence of the quotient can be understood through domain walls.

For a binary configuration x, define the wall variable

$$d_i = x_i \oplus x_{i+1},$$

where $\oplus$denotes addition modulo two.

A wall value

$$d_i = 1$$

indicates that neighboring cells differ. A block is homogeneous exactly when all of its internal wall variables vanish. It is mixed when at least one internal wall is present.

Applying Rule 178 to the cell variables induces a rule on the wall variables. Direct calculation gives

$$d_i' = d_i \vee (d_{i-1} \oplus d_{i+1}).$$

This is elementary Rule 222 under the standard encoding.

The wall formulation reveals two key properties.

First, an existing wall persists because

$$d_i = 1 \Longrightarrow d_i' = 1.$$

No update can remove a wall already present at the same position.

Second, new walls can be created at the boundaries of regions containing no walls. The expression

$$d_{i-1} \oplus d_{i+1}$$

detects asymmetric neighboring wall conditions. The boundary of a wall-free interval therefore advances inward.

Homogeneous regions correspond to intervals of zeros in the wall configuration. Under Rule 222, these intervals shrink from their boundaries unless the entire surrounding configuration is wall-free.

This mechanism explains why three homogeneous-zero blocks remain homogeneous zero and why three homogeneous-one blocks remain homogeneous one. In both cases there are no domain walls anywhere in the three-block input.

Every other class combination introduces either an internal wall within a mixed block or a boundary wall between homogeneous blocks of opposite values. The induced wall dynamics then determines whether at least one wall remains within the central output block.

The quotient theorem becomes a statement about the impossibility of clearing all central internal walls after the prescribed evolution time.

5.8 Proof of persistence at every block size greater than or equal to three

The computational search verifies the quotient at finite scales, but the domain-wall representation supports an all-scale result.

Consider three adjacent blocks of length k, evolved for $k$steps. Suppose the input is not uniformly $0^{3k}$and not uniformly 1^{3k}. Then the initial configuration contains at least one domain wall somewhere in the three-block region.

If a wall lies within the central block, it persists under Rule 222 and therefore remains evidence of mixedness unless it exits the central readout region. If walls lie outside the central block, the boundaries of the corresponding wall-free interval propagate inward.

For a central output block to be homogeneous, all of its $k-1$internal wall positions would have to be zero after $k$steps.

At block size two, there is only one internal wall position. Opposing fronts can arrange a stable pattern around this single position that allows the quotient condition to fail.

At block size three, there are two internal wall positions. The geometry changes qualitatively. A nonuniform three-block input cannot produce a wall-free interval covering both central internal positions after three steps unless the entire original three-block configuration was homogeneous.

For every $k \geq 3$, the central block contains at least two internal edges. Rule 222's persistent-wall and inward-front behavior prevents all of these edges from simultaneously becoming wall-free under a nonuniform input.

Therefore,

$$P_k\left(\Phi_{178,k}(u,v,w)\right) = m\left(P_k(u), P_k(v), P_k(w)\right)$$

for all

$$k \geq 3$$

and all

$$u, v, w \in B_k.$$

The result is an exact threshold theorem:

$$\boxed{P_k \text{ defines the semilattice-supporting quotient exactly when } k \geq 3.}$$

The computations through larger finite scales serve as independent checks, while the domain-wall argument explains why the pattern continues indefinitely.

5.9 Why the semilattice is not inherited from microscopic binary minors

The final issue is whether the semilattice represents genuine algebraic formation or merely a disguised microscopic operation.

At the cell level, the six binary minors of Rule 178 are projections. Each has one of the forms

$$p_L(x,y) = x$$

or

$$p_R(x, y) = y.$$

Neither is commutative on the two-element set:

$$p_L(0,1) = 0, p_L(1,0) = 1.$$

Similarly,

$$p_R(0,1) = 1, p_R(1,0) = 0.$$

The scale-three quotient operation satisfies

$$x \wedge y = y \wedge x.$$

Therefore the macro-operation cannot be isomorphic to any microscopic binary minor.

The difference is structural, not cosmetic. Microscopic projections preserve one designated argument and discard the other. The emergent semilattice treats its arguments symmetrically and combines their information through an exact meet.

At the microscopic level, argument position determines the result. At the quotient level, the result depends on agreement and conflict among macroclasses.

The relevant noninheritance statement is

$$\mathrm{Th}(\wedge) \subseteq/ \mathrm{Th}(p_L)$$

and

$$\mathrm{Th}(\wedge) \subseteq/ \mathrm{Th}(p_R),$$

because commutativity holds in the quotient semilattice but fails for both projections.

The result does not imply that the semilattice is logically independent of Rule 178. It is derived exactly from the rule. Rather, it shows that spatial aggregation and quotient formation create an operation with an equational theory unavailable among the corresponding microscopic minors.

Rule 178 therefore supplies a concrete example of strong Algebraic Abiogenesis:

asymmetric microscopic projection behavior ⟶ exact commutative semilattice at scale three.

The birth is thresholded, exact, persistent, and accompanied by a microscopic noninheritance certificate.

6. Contextual Transformation Monoids

6.1 From algebra on states to algebra on transformations

The Rule 178 construction studies an algebra whose elements are quotient states and whose operations act directly on those states. A more radical approach is to treat transformations themselves as the algebraic objects.

This shift replaces the question

Which operation do the states obey?

with

Which transformations can the dynamics generate, and what happens when those transformations are compos

A cellular automaton naturally supplies such transformations. At a fixed block scale, the future of one block depends on that block together with its surrounding blocks. If the surrounding blocks are held fixed while the central block varies, the cellular automaton induces a unary map on the block-state space.

Different fixed surroundings produce different maps. These maps can then be composed by taking the output of one context-controlled evolution as the input to another. Closure under composition creates a transformation monoid.

The emergent algebraic elements are therefore not the binary words themselves but functions

$$g: B_k \to B_k.$$

The algebraic operation is function composition:

$$(g, h) \mapsto g \circ h.$$

This construction is important for Algebraic Abiogenesis because the original cellular rule is not presented as a finite monoid of macrostate transformations. It is given only as a local ternary update. The monoid appears after block formation, context fixing, and compositional closure.

The resulting structure may contain:

- identity transformations;
- permutations;
- noninvertible maps;
- constant maps;
- idempotent maps;
- projections onto subsets;
- resets;
- cycles;
- and transformations implementing logical functions.

Thus a single local rule can generate a large algebra of controlled actions on higher-level states.

6.2 One-variable contexts of exact supercell operations

Let

$$\Phi_{F,k}: B_k^3 \to B_k$$

be the exact scale-k supercell operation of a cellular automaton F.

A one-variable context is obtained by fixing two arguments and allowing the remaining argument to vary.

There are three positional families.

For fixed $a, b \in B_k$, varying the left argument gives

$$L_{a,b}(x) = \Phi_{F,k}(x, a, b).$$

Varying the central argument gives

$$C_{a,b}(x) = \Phi_{F,k}(a, x, b).$$

Varying the right argument gives

$$R_{a,b}(x) = \Phi_{F,k}(a, b, x).$$

Each is a function

$$B_k \to B_k.$$

Since there are 2^kpossible fixed values for each of the two contextual blocks, each positional family contains at most

$$2^{2k}$$

maps. Across all three variable positions, there are at most

$$3 \cdot 2^{2k}$$

elementary context maps, though many may coincide.

At scale two,

$$| B_2 | = 4.$$

There are therefore at most

$$3 \cdot 4^2 = 48$$

elementary contexts. After duplicates are removed, the resulting generating set may be substantially smaller.

These contexts are exact controlled transformations. For example, a central-variable context

$$C_{a,b}(x)$$

means:

1. place fixed block $a$to the left of variable block x;
2. place fixed block $b$to the right;
3. evolve the three-block configuration for $k$steps;

4. read the central output block.

The fixed blocks act as a program or control environment. The variable block acts as the current macrostate.

This does not require that the context remain physically present through an autonomous long trajectory. The construction defines one exact controlled transformation. A new context may be supplied before the next transformation is applied.

The context maps are therefore analogous to elementary instructions available from the local dynamics under controlled boundary conditions.

6.3 Definition of the contextual transformation monoid

Let

$$\Lambda_k(F)$$

denote the set of all distinct elementary one-variable context maps derived from $\Phi_{F,k}$.

The contextual transformation monoid is

$$\boxed{\mathrm{C}_k(F) = \langle \Lambda_k(F) \rangle}$$

where the brackets denote closure under function composition and inclusion of the identity map.

Equivalently,

$$\mathrm{C}_k(F) \leq T_{2^k},$$

where T_n is the full transformation monoid on an n-element set.

The full transformation monoid contains every function

$$X \to X$$

for a set X of cardinality n. Its size is

$$| T_n | = n^n.$$

For block size k,

$$n = 2^k,$$

so

$$| T_{2^k} | = (2^k)^{2^k}.$$

At scale two,

$$| T_4 | = 4^4 = 256.$$

At scale three,

$$| T_8 | = 8^8 = 16{,}777{,}216.$$

The contextual monoid may be very small, containing only the identity and a few constant maps. It may be a group if every generated transformation is invertible. It may contain a mixture of permutations and singular transformations. In exceptional cases it may equal the entire transformation monoid.

The sequence

$$\mathfrak{C}(F) = (\mathsf{C}_1(F), \mathsf{C}_2(F), \mathsf{C}_3(F), \dots)$$

is the contextual monoid spectrum of the cellular automaton.

This spectrum records how the compositional transformation capacity of the rule changes with scale.

Two cellular automata with different microscopic lookup tables may possess isomorphic contextual monoids at one scale and diverge at another. The entire spectrum therefore provides a finer invariant than a single monoid.

6.4 Composition as the emergent algebraic operation

The local CA rule combines neighboring states through one update. The contextual monoid introduces a new operation not explicitly represented in the lookup table: composition of macro-transformations.

If

$$g, h \in \mathsf{C}_k(F),$$

then

$$g \circ h$$

is defined by

$$(g \circ h)(x) = g(h(x)).$$

Operationally, the variable block is first placed in the context implementing h. The resulting block is then placed in the context implementing g.

Composition is associative:

$$(f \circ g) \circ h = f \circ (g \circ h).$$

The identity transformation

$$\mathrm{id}_{B_k}(x) = x$$

acts as the monoid identity:

$$g \circ \mathrm{id} = \mathrm{id} \circ g = g.$$

These laws are not discoveries unique to the cellular automaton; they are universal properties of function composition. What the cellular automaton determines is the set of transformations available for composition and the closure they generate.

The emergent question is therefore not whether composition is associative. It is which submonoid of $T_{2^k}$the rule generates.

A small change in the elementary context maps can produce a dramatic change in closure. Two permutations may generate a large symmetric group. Adding one singular transformation of sufficiently high rank may expand the closure from a group to the full transformation monoid.

The higher algebra is therefore produced through iterative compositional amplification:

few elementary contexts ⟶ many composite transformations.

This resembles the generation of a group from a small set of permutations or the construction of a programming language from a limited instruction set.

A context word

$$w = g_1 g_2 \cdots g_r$$

represents a composite transformation. Depending on composition convention, its action is evaluated in a fixed order, for example

$$w(x) = g_r \circ \cdots \circ g_2 \circ g_1(x).$$

The word length $r$measures the number of controlled CA context applications required to synthesize the transformation.

Searching for short words that realize specified maps converts the contextual monoid into an exact finite program-synthesis environment.

6.5 Units, singular maps, ranks, and transformation structure

Every finite transformation monoid contains a group of units consisting of its invertible elements:

$$U(\mathrm{C}_k(F)) = \{g \in \mathrm{C}_k(F) : g \text{ is bijective}\}.$$

This is the contextual unit group, denoted

$$G_k(F) = U(\mathrm{C}_k(F)).$$

It acts as a permutation group on the block-state space B_k.

The unit group measures reversible contextual control. If

$$g \in G_k(F),$$

then there exists another generated transformation g^{-1}such that

$$g^{-1} \circ g = g \circ g^{-1} = \mathrm{id}.$$

Noninvertible transformations are singular. They merge states and therefore lose information.

The rank of a transformation is the size of its image:

$$\mathrm{rank}(g) = |\ g(B_k)\ |.$$

For a four-state block space, possible ranks are

$$1,2,3,4.$$

Rank-four maps are permutations. Rank-one maps are constants. Rank-two and rank-three maps perform intermediate forms of compression.

The rank distribution of a contextual monoid provides a concise structural profile:

$$r_j = |\ \{g \in \mathsf{C}_k(F) : \mathrm{rank}(g) = j\}\ |.$$

For the full transformation monoid T_4, the distribution is

Rank	Number of maps
1	4
2	84
3	144
4	24

The 24 rank-four maps form

$$S_4.$$

A contextual monoid equal to T_4therefore contains every reversible relabeling of four states, every possible irreversible compression, every constant reset, and every logical transformation of a two-bit block.

Idempotent transformations satisfy

$$e \circ e = e.$$

They act as stable projections, resets, or normalizations. Constant maps are idempotent, but nonconstant idempotents may project the state space onto a smaller invariant subset.

The interaction between units and singular maps is often decisive. A rich permutation group can move a singular map through many equivalent positions, producing a large family of transformations. In finite transformation theory, a sufficiently strong permutation group combined with one appropriate singular map can generate the full transformation monoid.

This mechanism explains how a small number of CA-derived contexts can produce complete macrostate programmability.

6.6 Contextual monoids as scale-indexed invariants

The contextual monoid depends on the scale:

$$\mathrm{C}_k(F).$$

The microscopic contextual monoid

$$\mathrm{C}_1(F)$$

acts on two states. Its unit group is necessarily a subgroup of

$$S_2,$$

and is therefore always Abelian.

At scale two, the contextual unit group acts on four states and can become non-Abelian. It may be isomorphic to groups such as

$$S_3, D_8, A_4, S_4.$$

This provides a precise sense in which non-Abelian symmetry can be born at a higher scale.

The scale spectrum is not generally monotone. It is possible that

$$| G_{k+1}(F) | < | G_k(F) |$$

even though the state space has grown. A rule may generate S_4at scale two but only a cyclic group or the trivial group at scale three.

Similarly,

$$\mathrm{C}_k(F) = T_{2^k}$$

does not imply

$$\mathrm{C}_{k+1}(F) = T_{2^{k+1}}.$$

Complete programmability at one block scale may disappear at another.

The scale-indexed monoid can therefore record:

- symmetry genesis;
- symmetry persistence;
- symmetry extinction;
- growth of irreversible control;
- changes in rank structure;
- and transitions between functional incompleteness and completeness.

A useful invariant is the sequence of monoid orders:

$$(|\ \mathsf{C}_1(F)\ |, |\ \mathsf{C}_2(F)\ |, |\ \mathsf{C}_3(F)\ |, \dots).$$

A stronger invariant records isomorphism types, unit groups, rank distributions, Green's relations, ideals, and minimal generating sets.

The entire contextual spectrum may distinguish rules that appear identical under simpler classifications. Two rules can share the same Wolfram class, entropy profile, or scale-two unit group while having different scale-three transformation structures.

Contextual monoids therefore provide a bridge between cellular-automaton dynamics, finite semigroup theory, and scale-dependent computation.

6.7 Relation to translation monoids and universal algebra

In universal algebra, fixing all but one argument of a fundamental operation produces elementary translations. Closure of these maps under composition gives the translation monoid of the algebra.

The contextual monoid is therefore naturally interpreted as the translation monoid of the exact block algebra:

$$\mathsf{C}_k(F) = \mathrm{Trans}(\mathcal{B}_k(F)).$$

This identifies the construction with an established algebraic concept. The novelty does not lie in inventing transformation monoids or translation monoids. It lies in applying them systematically to exact scale-indexed CA block algebras and treating their changing structure as a form of algebraic emergence.

Universal algebra usually begins with a supplied finite algebra and studies its terms, congruences, subalgebras, and translations. Here the finite algebra is itself derived from a lower-level dynamical substrate.

The full chain is

$$F \longrightarrow \mathcal{B}_k(F) \longrightarrow \mathrm{Trans}(\mathcal{B}_k(F)).$$

This creates a two-stage formation process.

First, local dynamics generates the block operation.

Second, the block operation generates a transformation monoid through contextual substitution and composition.

The construction can be extended beyond elementary cellular automata. Any finite-radius CA over a finite alphabet produces finite block algebras. Multistate systems lead to transformation monoids on q^kstates. Higher-dimensional systems produce operations with more contextual arguments.

The translation-monoid perspective also interacts with congruence theory. A congruence of the block algebra must be preserved by every translation. Therefore the contextual monoid acts on the lattice of admissible quotient policies.

Conversely, a quotient homomorphism induces a homomorphic image of the contextual monoid. One may study how transformation capacity changes under observational compression:

$$\mathsf{C}_k(F) \longrightarrow \mathsf{C}_k(F)/\theta.$$

This joins the state-based and transformation-based branches of Algebraic Abiogenesis.

6.8 Controlled context composition versus autonomous CA evolution

The contextual-monoid construction requires an important interpretive boundary.

A composite transformation

$$g_r \circ \cdots \circ g_1$$

is implemented by applying a sequence of controlled environments. Before each step, the surrounding blocks associated with the next context may be reset or supplied externally.

This differs from placing one finite initial configuration into an infinite CA lattice and allowing it to evolve autonomously without intervention.

Therefore,

$$\mathrm{C}_k(F) = T_{2^k}$$

does not imply that every transformation is spontaneously executed by one isolated CA trajectory.

It means that the local rule, when used as a context-programmable primitive, can synthesize every macrostate transformation through finite controlled compositions.

The distinction is analogous to the difference between a transistor's physical dynamics and a computer assembled from transistors under engineered wiring and clocking. The transistor does not autonomously produce every program. Its controlled compositional use makes universal computation possible.

In the CA setting, the contexts play the role of external controls or program instructions. The block being transformed plays the role of a register.

This form of universality may be called contextual functional completeness:

$$\boxed{\mathrm{C}_k(F) = T_{2^k}.}$$

It is stronger than possessing one universal Boolean gate on a selected encoding because it includes every function on the block-state space. It is weaker than autonomous Turing universality because it assumes context selection and staged composition.

The distinction should be maintained throughout the paper:

$$\begin{aligned} \text{Autonomous dynamics:} &\quad x \mapsto F^t(x), \\ \text{Contextual control:} &\quad x \mapsto g_r \circ \cdots \circ g_1(x). \end{aligned}$$

Both are legitimate mathematical systems, but they answer different questions.

For Algebraic Abiogenesis, contextual control remains important because it demonstrates that the algebra of available transformations can be far richer than the native microscopic update suggests. A small local rule can become, under block formation and contextual composition, a complete algebra of macrostate manipulation.

7. Rule 30, Full Transformation Completeness, and NAND

7.1 The two-cell block state space

Rule 30 is one of the best-known elementary cellular automata because a short asymmetric lookup table produces irregular and apparently complex spacetime behavior. Its local rule acts on binary neighborhoods according to

$$111 \mapsto 0,110 \mapsto 0,101 \mapsto 0,100 \mapsto 1,$$
$$011 \mapsto 1,010 \mapsto 1,001 \mapsto 1,000 \mapsto 0.$$

For the present analysis, the visual complexity of Rule 30 is secondary. The relevant object is its exact scale-two block algebra.

At block size two,

$$B_2 = \{00,01,10,11\}.$$

For convenience, encode these four blocks as

$$0 = 00,1 = 01,2 = 10,3 = 11.$$

The scale-two supercell operation

$$\Phi_{30,2}: B_2^3 \to B_2$$

is obtained by concatenating three two-cell blocks, evolving the resulting six-cell configuration for two Rule 30 steps, and reading the central two output cells.

The operation therefore has

$$4^3 = 64$$

entries.

Fixing two of its arguments and varying the third produces unary transformations of the four-state block space. The complete family contains left-variable, central-variable, and right-variable contexts.

Each context is a map

$$g: B_2 \to B_2.$$

Under the numerical encoding, such a map can be represented by a four-tuple

$$(g(0), g(1), g(2), g(3)).$$

For example,

$$g = (3,0,1,2)$$

means

$$0 \mapsto 3,1 \mapsto 0,2 \mapsto 1,3 \mapsto 2.$$

The question is not merely how many distinct elementary contexts exist. It is what transformations can be generated by composing them.

The full set of all transformations on four states is

$$T_4,$$

with cardinality

$$| T_4 |= 4^4 = 256.$$

The full group of invertible transformations is

$$S_4,$$

with cardinality

$$| S_4 |= 4! = 24.$$

The central computational result is that Rule 30 reaches both maxima at scale two.

7.2 Exact generation of the full transformation monoid T_4

Let

$$\Lambda_2(30)$$

be the set of all distinct elementary one-variable contexts of the scale-two Rule 30 block operation.

The contextual transformation monoid is

$$\mathrm{C}_2(30) = \langle \Lambda_2(30) \rangle.$$

Exact closure under composition gives

$$\boxed{\mathrm{C}_2(30) = T_4.}$$

Thus every function

$$B_2 \to B_2$$

can be synthesized as a finite composition of Rule 30 contexts.

This includes all:

24 permutations,
144 rank-three maps,
84 rank-two maps,
4 constant maps.

The rank distribution is exactly that of the full transformation monoid:

Rank	Number of transformations
1	4
2	84
3	144
4	24

No transformation of the four-state block space lies outside the generated closure.

This result is stronger than the observation that Rule 30 can implement one logical gate. It establishes complete finite-state controllability at the chosen block scale under context resetting.

For every target map

$$h: B_2 \to B_2,$$

there exists a finite word

$$g_r \cdots g_2 g_1$$

in elementary Rule 30 contexts such that

$$h = g_r \circ \cdots \circ g_2 \circ g_1.$$

The generated algebra therefore contains every reversible operation, every irreversible compression, every reset, every state merger, and every Boolean transformation compatible with the four-state encoding.

The result is exact and finite. It follows from exhaustive closure, not from statistical sampling or visual inspection.

7.3 A minimal three-context generating certificate

Although the complete elementary context family generates T_4, a much smaller certificate suffices.

Three Rule 30 context maps are

$$p = (3,0,1,2),$$
$$q = (3,2,0,1),$$

and

$$r = (0,1,3,3).$$

The maps $p$and $q$are permutations. Their generated subgroup has order 24:

$$\langle p, q \rangle = S_4.$$

The map $r$is singular because two inputs are mapped to the same output. Its image is

$$\{0,1,3\},$$

so

$$\mathrm{rank}(r) = 3.$$

Adding this one rank-three transformation to the symmetric group gives

$$\boxed{\langle p, q, r\rangle = T_4.}$$

The certificate is minimal in number of generators for this standard transformation-theoretic reason.

A generating set for T_4must contain at least one singular map, because permutations alone generate only a group. It must also contain enough invertible maps to generate S_4. Since $S_4$is not cyclic, at least two permutation generators are required.

Therefore any generating certificate of this form requires at least

$$2 + 1 = 3$$

maps.

The triple (p, q, r)achieves that lower bound.

The significance is not merely compression of the computational output. The three-map certificate shows that the full 256-element monoid does not require a large collection of unrelated contexts. A minimal controlled instruction set already exists within Rule 30's scale-two block dynamics.

The three maps play distinct roles:

p, q : reversible rearrangement,
r : irreversible state identification.

Together they generate the entire finite transformation universe on four macrostates.

7.4 Emergence of the non-Abelian symmetric group S_4

At the microscopic scale, Rule 30 acts on the two-element set

$$B_1 = \{0,1\}.$$

Any invertible unary transformation on two states belongs to

$$S_2 \cong C_2,$$

which is Abelian.

At scale two, the contextual unit group is

$$G_2(30) = U(\mathrm{C}_2(30)) = S_4.$$

This group is non-Abelian:

$$p \circ q \neq q \circ p.$$

The transition from scale one to scale two therefore includes a precise symmetry-birth event:

$$C_2 \longrightarrow S_4.$$

The non-Abelian structure is not introduced as an external target. It results from the permutations generated by controlled block contexts.

The appearance of S_4has several consequences. The context system can send any block state to any other block state. It can realize every permutation of the four macrostates. It can implement transpositions, cycles, double transpositions, and all conjugate variants of these transformations.

The group acts transitively on B_2, and more strongly, it realizes every possible relabeling of the four block states.

This means that no block state is algebraically privileged under the reversible context action. Any state can be moved to any selected position in the four-state space.

The transition is an example of weak Algebraic Abiogenesis or algebraic morphogenesis. The microscopic rule already acts on binary states and contains nontrivial Boolean structure. However, the full non-Abelian group $S_4$is not explicitly presented in the local lookup table. It becomes available only after spatial aggregation and compositional closure.

The group is therefore an emergent organization of transformations rather than an emergent binary law on states.

7.5 Exact two-context synthesis of embedded NAND

The four two-cell blocks can be interpreted as ordered pairs of bits:

$$00,01,10,11.$$

Consider the transformation

$$N(x,y) = (x, \mathrm{NAND}(x,y)).$$

Since

$$\mathrm{NAND}(x,y) = 1 - xy$$

on binary inputs, the map acts as follows:

$$00 \mapsto 01,$$
$$01 \mapsto 01,$$
$$10 \mapsto 11,$$
$$11 \mapsto 10.$$

Under the numerical block encoding, this is

$$N = (1,1,3,2).$$

Because

$$N \in T_4 = \mathrm{C}_2(30),$$

it must have a context-word representation.

An exact shortest synthesis over the full elementary context set uses two contexts. Define

$$a = (2,3,1,0)$$

and

$$b = (2,3,1,1).$$

Then

$$\boxed{b \circ a = (1,1,3,2).}$$

Thus

$$b \circ a$$

implements the embedded NAND transformation.

The first context rearranges the four block states. The second performs the required irreversible identification and output assignment. Neither context alone is the NAND map.

The synthesis is globally shortest because no single elementary context has the target table, while a word of length two does.

This correction is important. A three-factor word exists within the selected minimal T_4generating set, but the shortest word over the complete elementary Rule 30 context set has length two.

The exact result is therefore:

Rule 30 synthesizes embedded NAND in two controlled scale-two context applications.

7.6 Functional completeness at the macrostate level

NAND is functionally complete for Boolean logic. Any Boolean function can be constructed from compositions of NAND gates when fan-out and wiring are available.

However, the contextual-monoid result is stronger in one finite-state sense and more limited in another.

It is stronger because

$$\mathrm{C}_2(30) = T_4$$

contains every transformation on the two-bit block space, not only transformations generated by a fixed Boolean gate syntax.

For example, the monoid contains:

bitwise complement maps,
bit swaps,
controlled bit changes,
constants,
erasures,
state-dependent resets,
all permutations,
all two-bit lookup functions.

It is more limited because the construction concerns unary transformations of one four-state register. It does not by itself provide an unbounded supply of registers, autonomous signal routing, persistent memory, or an internally generated sequence of control contexts.

The correct statement is therefore contextual functional completeness:

$$\boxed{\mathrm{C}_2(F) = T_4.}$$

This means that the rule is complete for controlled transformations of one two-cell macrostate.

It does not immediately imply Turing universality. Turing universality concerns the ability of an autonomous or finitely programmed system to simulate arbitrary computation with potentially unbounded memory and time.

Contextual completeness concerns exact finite control over a fixed macrostate space under externally selected contexts.

The two notions are related but not equivalent.

The Rule 30 result nevertheless provides a direct bridge to the original NAND-oriented ambition of Algebraic Abiogenesis. The system was not seeded with a NAND circuit identity. It first constructed the complete transformation monoid and only afterward recognized that NAND was one element of that monoid.

The discovery sequence was therefore:

$$\text{anonymous rule} \longrightarrow \text{block contexts} \longrightarrow T_4 \longrightarrow \text{NAND identification.}$$

This is preferable to asking the system from the beginning to synthesize a known NAND gate.

7.7 The fourteen elementary rules generating T_4

The Rule 30 result is not isolated. Exhaustive analysis of all 256 elementary cellular automata at block size two identified exactly 14 rules satisfying

$$\mathrm{C}_2(F) = T_4.$$

They are

$$30,\ 73,\ 86,\ 106,\ 109,\ 110,\ 120,$$
$$124,\ 135,\ 137,\ 149,\ 169,\ 193,\ 225.$$

Every one of these rules generates all 256 transformations of the four two-cell blocks.

The classification is complete for elementary cellular automata under the exact scale-two construction. Every rule was processed using the same policy. No rule-specific context selection was introduced.

For each rule:

1. the 64-entry supercell operation was computed;
2. every elementary one-variable context was extracted;
3. duplicate maps were removed;
4. closure under composition was calculated;
5. the resulting monoid was compared with T_4.

The 14 positive rules therefore form a precise class of scale-two context-complete elementary automata.

This class cuts across familiar behavioral labels. It contains Rule 30, commonly associated with irregular dynamics, and Rule 110, associated with autonomous computational universality. It also contains rules less prominent in standard CA discussions.

The result suggests that contextual transformation completeness measures a property different from visual complexity, entropy, particle richness, or known Turing universality.

A rule can possess full finite macrostate controllability without being conventionally identified as computationally universal. Conversely, a rule may exhibit complicated spacetime behavior without generating the full transformation monoid at the chosen block scale.

7.8 Symmetry classes and equivalence among the fourteen rules

Elementary cellular automata are related by reflection and state complementation. These symmetries transform lookup tables while preserving many structural properties.

The 14 context-complete rules form four reflection-complement classes:

$$\{30,86,135,149\},$$
$$\{73,109\},$$
$$\{106,120,169,225\},$$

and

$$\{110,124,137,193\}.$$

Rules within one class are related by reversing the spatial direction, exchanging zero and one, or combining these transformations.

Their contextual monoids are isomorphic because the symmetry transformation induces a relabeling of block states and contexts.

The 14-rule count therefore corresponds to four essentially distinct microscopic rule families.

This reduction is useful when interpreting the classification. The property

$$\mathsf{C}_2(F) = T_4$$

is not randomly distributed across 14 unrelated rules. It occurs in four structural families and all their elementary symmetry variants.

However, equivalence at scale two does not imply equivalence at larger scales.

The four families diverge at scale three. Some retain large non-Abelian contextual unit groups. Others collapse to small cyclic groups or even the trivial group.

Thus the scale-two classification is real but incomplete as a global classification of cellular automata.

The appropriate invariant is not merely

$$\mathsf{C}_2(F),$$

but the entire sequence

$$\mathfrak{C}(F) = (\mathsf{C}_1(F), \mathsf{C}_2(F), \mathsf{C}_3(F), \dots).$$

The 14 rules agree at one scale and separate at the next. Their common T_4behavior is therefore a scale-local universality class.

7.9 What contextual functional completeness does and does not imply

The equality

$$\mathsf{C}_2(30) = T_4$$

supports several strong conclusions.

It implies that every transformation of the four two-cell block states can be synthesized exactly through controlled Rule 30 contexts.

It implies that the reversible transformations form the full non-Abelian group

$$S_4.$$

It implies that constant resets, every possible state merger, and every rank profile are present.

It implies that embedded Boolean transformations such as NAND can be synthesized.

It does not imply that an uncontrolled Rule 30 spacetime trajectory will spontaneously perform an arbitrary selected transformation.

It does not imply that Rule 30 autonomously supplies the environmental resets used between context applications.

It does not imply that the contextual synthesis automatically scales to unbounded memory.

It does not prove Turing universality.

It does not establish that NAND is a novel Boolean operation. NAND is used as a recognizable element of the discovered transformation monoid.

The scientific importance lies in the mode of derivation. The complete transformation algebra is obtained from an anonymous local rule through exact block formation and context composition. The target monoid and NAND map need not be supplied during the discovery stage.

Rule 30 therefore demonstrates a form of operation-space birth:

$$\boxed{\text{local binary dynamics} \longrightarrow \text{complete macrostate transformation algebra.}}$$

The result extends Algebraic Abiogenesis beyond the formation of laws on quotient states. It shows that a lower-level dynamical substrate can also generate a complete algebra of controlled actions, including reversible symmetry, irreversible compression, and universal finite-state logic.

8. Contextual Symmetry Spectra Across Scale

8.1 The unit group of a contextual transformation monoid

For each cellular automaton $F$and block scale k, the contextual transformation monoid

$$\mathrm{C}_k(F)$$

contains all unary block transformations obtainable by composing exact one-variable supercell contexts. Some of these transformations are irreversible. They merge states, erase distinctions, or collapse the block space onto a smaller image. Others are bijective and therefore reversible.

The invertible elements form the contextual unit group

$$G_k(F) = U(\mathrm{C}_k(F)).$$

Since $G_k(F)$acts by permutations on the block state space

$$B_k = \{0,1\}^k,$$

it is a subgroup of

$$S_{2^k}.$$

The group $G_k(F)$describes the reversible symmetries made available by controlled contexts at scale k. It is not necessarily the automorphism group of the block algebra. An automorphism must preserve the original supercell operation, whereas a contextual unit is generated through translations and their compositions. The two constructions answer different questions.

The automorphism group asks which relabelings leave the algebra unchanged. The contextual unit group asks which reversible state transformations the algebra itself can perform when embedded into admissible contexts.

At scale one, the state space has two elements. Therefore

$$G_1(F) \leq S_2,$$

and every microscopic contextual unit group is either trivial or cyclic of order two. In particular, no elementary cellular automaton can possess a non-Abelian contextual unit group at the one-cell scale.

At scale two, the block space has four states, so

$$G_2(F) \leq S_4.$$

Non-Abelian groups can now occur. At scale three,

$$G_3(F) \leq S_8,$$

allowing substantially richer reversible organization.

The birth of non-Abelian contextual symmetry is therefore impossible microscopically but possible after block formation. This creates an exact algebraic threshold between single-cell and multi-cell observation.

8.2 Definition of the contextual symmetry spectrum

A single contextual unit group gives only a scale-local description. The more informative object is the full sequence

$$\boxed{\mathfrak{G}(F) = (G_1(F), G_2(F), G_3(F), \dots).}$$

This sequence is the contextual symmetry spectrum of F.

The spectrum records the reversible transformation structure available at every block scale under the exact supercell-context policy. It may be represented by group isomorphism types, group orders, generating sets, orbit structures, solvability classes, derived lengths, transitivity properties, or actions on block states.

A minimal numerical spectrum records

$$(|\ G_1(F)\ |, |\ G_2(F)\ |, |\ G_3(F)\ |, \dots).$$

A more complete spectrum records

$$([G_1(F)], [G_2(F)], [G_3(F)], \dots),$$

where $[G]$denotes the isomorphism class of the group.

The sequence is not generally equipped with canonical inclusion maps. The block spaces differ:

$$|\ B_k\ | = 2^k, |\ B_{k+1}\ | = 2^{k+1}.$$

A transformation at scale $k$therefore does not automatically embed into the transformation group at scale $k+1$. Comparisons between scales must be made through explicit block encodings, residue-class decompositions, quotient maps, or other structural constructions.

The spectrum is consequently not a simple filtration. It is a family of distinct groups generated from one microscopic rule under different observation policies.

This allows two cellular automata to agree at one scale and diverge at another:

$$G_2(F) \cong G_2(H),$$

while

$$G_3(F) \ncong G_3(H).$$

Their scale-two symmetry is the same, but their symmetry spectra are different.

The entire sequence can therefore serve as a dynamical-algebraic fingerprint.

8.3 Complete elementary-rule classification at scales two and three

The contextual unit groups were computed for all 256 elementary cellular automata at scales two and three.

At scale two, exactly eight group isomorphism types occur:

Group type	Number of rules
1	154
C_2	48
$C_2 \times C_2$	18
C_4	4
S_3	2
D_8	12
A_4	4
S_4	14

Here 1denotes the trivial group, $C_n$the cyclic group of order n, $D_8$the dihedral group of order eight, and $A_4$and $S_4$the alternating and symmetric groups on four points.

The distribution shows that most elementary rules possess little reversible contextual control at scale two. More than half have no nontrivial contextual unit. Nevertheless, a small subset produces large non-Abelian groups, including the full symmetric group S_4.

At scale three, exactly six group types occur:

Group type	Order	Number of rules
1	1	221
C_2	2	7
C_4	4	4
$C_2 \times S_4$	48	2
$C_2 \wr C_2 \wr C_2$	128	16
$AGL(3,2)$	1344	6

The group

$$C_2 \wr C_2 \wr C_2$$

is an iterated wreath product of order 128 and may be identified with a Sylow two-subgroup of S_8.

The affine general linear group

$$AGL(3,2) = \mathbb{F}_2^3 \rtimes GL(3,2)$$

has order

$$8 \cdot 168 = 1344.$$

No elementary rule generates the full symmetric group

$$S_8$$

as its scale-three contextual unit group under the tested policy.

The scale-three distribution is more concentrated than the scale-two distribution. Most rules become contextually rigid, while a small collection produces very large affine or two-group symmetries.

This classification is exact for the finite rule space and the stated block policy.

8.4 Symmetry genesis between adjacent scales

Symmetry genesis occurs when a group property absent at scale $k$appears at scale $k + 1$.

For example, non-Abelian symmetry is born when

$$G_k(F) \text{ is Abelian}$$

but

$$G_{k+1}(F) \text{ is non-Abelian.}$$

Since every subgroup of $S_2$is Abelian, any rule with non-Abelian $G_2(F)$exhibits non-Abelian symmetry genesis between scales one and two.

At the transition from scale two to scale three, eight elementary rules acquire non-Abelian symmetry after possessing only Abelian symmetry at scale two.

This demonstrates that block aggregation can expose reversible relations that were unavailable at the preceding scale. The additional block states do not merely enlarge the permutation domain. They permit context maps whose generated compositions fail to commute.

The birth event can be expressed through the commutator subgroup:

$$[G_k(F), G_k(F)] = 1,$$

but

$$[G_{k+1}(F), G_{k+1}(F)] \neq 1.$$

Other symmetry properties can be treated similarly. Transitivity, primitivity, solvability, nilpotence, and simplicity may each appear at particular scales.

A general symmetry-genesis event is therefore a transition

$$P(G_k(F)) = \text{false}, P(G_{k+1}(F)) = \text{true},$$

for a group-theoretic property P.

The contextual symmetry spectrum converts qualitative emergence into a sequence of exact algebraic phase changes.

8.5 Persistence of non-Abelian organization

Symmetry persistence occurs when a structural property survives a change of scale.

Sixteen elementary rules possess non-Abelian contextual unit groups at both scales two and three.

This does not require the groups themselves to remain isomorphic. A rule may move from

$$S_4$$

at scale two to an order-128 wreath-product group at scale three. The specific symmetry algebra changes, but noncommutative reversible organization persists.

Persistence may therefore be defined at several levels.

Exact group persistence requires

$$G_k(F) \cong G_{k+1}(F).$$

Property persistence requires only

$$P(G_k(F)) \text{and} P(G_{k+1}(F)).$$

Core persistence may involve the preservation of a quotient, factor, or action even when the full group changes.

The additive cellular automata later provide a stronger arithmetic form of persistence. Along scales related by multiplication by the field characteristic, the nontranslation symmetry core remains fixed while the translation module expands.

Thus the appropriate scale relation need not be adjacency. A group property may persist along an arithmetic ray

$$m, 2m, 4m, 8m, \ldots$$

even while it changes irregularly between consecutive integers.

This distinction becomes essential in interpreting apparently erratic spectra. What looks like instability under $k \mapsto k + 1$may conceal exact stability under $k \mapsto 2k$.

8.6 Symmetry extinction under increased block size

Symmetry extinction occurs when a property present at one scale disappears at a larger scale.

Sixteen elementary rules have non-Abelian contextual unit groups at scale two but Abelian unit groups at scale three.

For such a rule,

$$[G_2(F), G_2(F)] \neq 1,$$

while

$$[G_3(F), G_3(F)] = 1.$$

This result contradicts the intuition that increasing block size necessarily preserves or enriches existing symmetry.

The larger block algebra contains more states, but its elementary context maps arise from a different exact supercell operation. The scale-three maps need not include lifted versions of the scale-two maps. The new causal geometry may eliminate the reversible mechanisms that generated the smaller-scale group.

Extinction can be even more severe. A rule may possess a large non-Abelian group at one scale and become contextually rigid at the next:

$$G_k(F) \neq 1, G_{k+1}(F) = 1.$$

This does not mean that the cellular automaton has lost all dynamical complexity. It means that under the particular scale-$(k+1)$ context policy, no nontrivial reversible block transformation is generated.

Symmetry extinction is therefore policy-relative. Another quotient, temporal depth, or block geometry could reveal a different group.

Nevertheless, within the fixed experimental protocol, extinction is exact and reproducible. It is a genuine change in the available transformation algebra.

8.7 The scale-three bifurcation of the T_4 rules

The fourteen rules satisfying

$$\mathsf{C}_2(F) = T_4$$

all have

$$G_2(F) = S_4.$$

At scale two, they are indistinguishable by their contextual monoids. Each can generate every transformation of the four block states.

At scale three, they split into three sharply different families.

Eight rules generate the order-128 wreath-product group:

$$30,\ 86,\ 106,\ 120,\ 135,\ 149,\ 169,\ 225.$$

For these rules,

$$G_3(F) \cong C_2 \wr C_2 \wr C_2.$$

Four rules generate a cyclic group of order four:

$$110,\ 124,\ 137,\ 193.$$

For these rules,

$$G_3(F) \cong C_4.$$

Two rules lose all nontrivial contextual units:

$$73,\ 109.$$

For them,

$$G_3(F) = 1.$$

Thus one scale-two universality class splits immediately into:

large non-Abelian symmetry,
small Abelian symmetry,

and

no reversible symmetry.

This bifurcation demonstrates why a single-scale classification is insufficient.

Rule 30 and Rule 110 both generate T_4at scale two. Yet at scale three, Rule 30 belongs to the order-128 family, while Rule 110 collapses to C_4.

Their contextual transformation capacities are identical at one resolution and radically different at the next.

The phenomenon resembles a degeneracy broken by a finer measurement. The scale-two invariant identifies a common capability, while the scale-three invariant reveals distinct underlying organizational mechanisms.

8.8 Failure of simple monotonicity in algebraic complexity

Several natural monotonicity conjectures fail.

Larger state space does not guarantee a larger unit group:

$$| G_{k+1}(F) | \not\geq | G_k(F) |.$$

Non-Abelian symmetry does not necessarily persist:

$$G_k(F) \text{ non-Abelian} \not\Longrightarrow G_{k+1}(F) \text{ non-Abelian}.$$

Abelian symmetry does not prevent later non-Abelian emergence:

$$G_k(F) \text{ Abelian} \not\Longrightarrow G_{k+1}(F) \text{ Abelian}.$$

Full transformation completeness at one scale does not imply completeness at the next:

$$\mathrm{C}_k(F) = T_{2^k} \not\Longrightarrow \mathrm{C}_{k+1}(F) = T_{2^{k+1}}.$$

The failure of monotonicity arises because the algebras at different scales are not related by simple containment. Each block scale defines a new operation through a new spacetime construction.

A larger block may reveal more internal degrees of freedom, but it may also impose stronger constraints on which global block permutations can be realized through exact contexts.

Group order alone is also an incomplete measure of algebraic complexity. A large solvable two-group and a smaller nonsolvable group may represent different kinds of organization. Transitivity, primitive action, derived length, composition factors, and normal-subgroup structure may be more informative than size.

The scale spectrum should therefore be treated as a sequence of algebraic phases rather than a steadily increasing hierarchy.

8.9 Algebraic universality classes based on scale spectra

A scale-local universality class groups rules whose contextual algebras are isomorphic at one chosen scale:

$$F \sim_k H \Leftrightarrow G_k(F) \cong G_k(H).$$

A stronger relation compares a finite prefix of the spectrum:

$$F \sim_{\leq n} H$$

when

$$G_k(F) \cong G_k(H)$$

for every

$$1 \leq k \leq n.$$

The strongest relation compares the complete contextual symmetry spectra:

$$F \sim_{\mathfrak{G}} H \Leftrightarrow \mathfrak{G}(F) \cong \mathfrak{G}(H)$$

under an appropriate scale-preserving notion of equivalence.

Such classifications can identify rules that differ microscopically but generate the same higher transformation organization.

Reflection and state complementation naturally produce equivalent spectra up to relabeling. More interestingly, unrelated microscopic rules may also converge to identical contextual group sequences.

Conversely, rules assigned to the same traditional behavioral category may possess different algebraic spectra.

The spectrum therefore offers a route toward algebraic universality classes based on exact transformation capacity rather than visual appearance.

A refined classification could record both monoids and groups:

$$(\mathrm{C}_k(F), G_k(F))_{k\geq 1}.$$

It could also include congruence lattices, quotient theories, and emergent term identities:

$$\mathfrak{E}_k(F) = (\mathrm{C}_k(F), G_k(F), \mathrm{Con}\,\mathcal{B}_k(F), \mathrm{Th}\,\mathcal{B}_k(F)).$$

The resulting invariant would describe an algebraic ecology at every scale.

The central conclusion is that a cellular automaton is not adequately characterized by one emergent symmetry group. Its mathematical identity unfolds across a scale-indexed spectrum in which symmetries may arise, persist, transform, divide into distinct families, or disappear.

9. Frobenius Towers and Arithmetic Selection of Symmetry

9.1 Additive cellular automata over finite fields

A special but mathematically important class of cellular automata is formed by additive rules over finite fields. In these systems, the next state of each cell is a linear combination of neighboring states.

For a one-dimensional cellular automaton over the field

$$\mathbb{F}_p,$$

the local rule may be written as

$$x_i(t+1) = \sum_{j=-r}^{r} a_j\, x_{i+j}(t),$$

where

$$a_j \in \mathbb{F}_p.$$

For binary additive elementary cellular automata,

$$p = 2$$

and

$$r = 1.$$

The update rule has the form

$$x_i(t+1) = ax_{i-1}(t) + bx_i(t) + cx_{i+1}(t) (\mathrm{mod} 2),$$

with

$$a, b, c \in \mathbb{F}_2.$$

There are eight such linear elementary rules, corresponding to the eight possible choices of the coefficient triple (a, b, c). Their complements form eight additional affine rules.

Additive cellular automata differ from nonlinear rules such as Rule 30 and Rule 178 because their microscopic dynamics already possess an explicit vector-space structure. Configurations can be added pointwise, and evolution preserves that addition:

$$F(x+y) = F(x) + F(y).$$

This means that their higher algebraic structures should not be interpreted as emerging from a substrate devoid of algebra. Instead, they provide a controlled setting in which elementary microscopic linearity reorganizes into larger scale-dependent groups.

The relevant form of emergence is algebraic morphogenesis:

local linear rule ⟶ block affine transformations ⟶ large contextual symmetry groups.

Because the microscopic rule is linear, the induced block maps can be represented by matrices. This makes it possible to derive exact all-scale formulas rather than relying only on finite enumeration.

Additive cellular automata therefore serve as the analytically tractable branch of Cellular Algebraic Abiogenesis. Nonlinear rules reveal unexpected algebraic birth, while additive rules reveal the arithmetic laws governing algebraic organization across scale.

9.2 Laurent-polynomial representations of local dynamics

A one-dimensional linear cellular automaton can be represented by a Laurent polynomial.

Let

$$P(z) = \sum_{j=-r}^{r} a_j z^j$$

over the field $\mathbb{F}_p$. The variable $z$represents spatial translation. Multiplication by $z$shifts a configuration by one lattice position, while multiplication by z^{-1}shifts it in the opposite direction.

A configuration with finite support may be represented as

$$X(z) = \sum_i x_i \, z^i.$$

One cellular-automaton update becomes

$$X_{t+1}(z) = P(z)X_t(z).$$

After $t$steps,

$$X_t(z) = P(z)^t X_0(z).$$

Thus the complete spacetime evolution is encoded in powers of the local rule polynomial.

For example, a rule with local update

$$x_i(t+1) = x_{i-1}(t) + x_i(t)$$

has polynomial

$$P(z) = z^{-1} + 1.$$

A rule with update

$$x_i(t+1) = x_{i-1}(t) + x_{i+1}(t)$$

has polynomial

$$P(z) = z^{-1} + z.$$

The coefficients of

$$P(z)^k$$

determine which initial cells influence each cell after $k$steps. They therefore determine the linear part of the exact scale-k supercell operation.

The block construction transforms the Laurent-polynomial evolution into finite matrices. The left, central, and right input blocks contribute through matrices

$$A_k, B_k, C_k,$$

so that

$$\Phi_{F,k}(u, v, w) = A_k u + B_k v + C_k w.$$

For affine rules, an additional constant vector may appear:

$$\Phi_{F,k}(u, v, w) = A_k u + B_k v + C_k w + d_k.$$

Fixing two arguments produces affine unary maps on

$$\mathbb{F}_p^k.$$

The contextual unit group is therefore an affine permutation group generated by the invertible coefficient matrices and the available translations.

The Laurent-polynomial representation makes the dependence on block size explicit. The same exponent $k$determines both the number of evolution steps and the scale of the central block. Arithmetic properties of $k$consequently enter directly through the expansion of

$$P(z)^k.$$

9.3 Frobenius decomposition under characteristic-p scale multiplication

The central all-scale mechanism follows from the Frobenius identity in characteristic p:

$$(a + b)^p = a^p + b^p.$$

More generally,

$$\left(\sum_i a_i\right)^{p^s} = \sum_i a_i^{p^s}.$$

Since coefficients in $\mathbb{F}_p$satisfy

$$a_i^{p^s} = a_i,$$

a Laurent polynomial obeys

$$P(z)^{p^s} = P\left(z^{p^s}\right).$$

If the block size is written as

$$k = p^s m, p \nmid m,$$

then

$$P(z)^k = (\quad(\quad)^m = P\left(z^{p^s}\right)^m.$$

This identity is the algebraic source of the contextual symmetry towers.

The substitution

$$z \mapsto z^{p^s}$$

means that only spatial offsets divisible by p^sappear in the expanded polynomial. The lattice decomposes into residue classes modulo p^s:

$$0,1, \ldots, p^s - 1.$$

Within each residue class, the evolution is governed by the same odd-core or characteristic-free dynamics associated with the scale m.

Thus the scale-$p^s m$ evolution splits into

$$p^s$$

copies of a scale-m evolution.

For binary additive cellular automata,

$$p = 2.$$

Writing

$$k = 2^s m$$

with $m$odd gives

$$P(z)^k = P\left(z^{2^s}\right)^m.$$

The power of two in the block size determines how many residue-class copies occur. The odd part determines the internal linear symmetry acting identically on those copies.

This decomposition explains why scales sharing the same odd part often have the same nontranslation symmetry core.

9.4 Residue-class splitting of supercell coefficient matrices

The Frobenius polynomial identity has a direct matrix consequence.

At scale

$$k = p^s m,$$

reorder the $k$block coordinates by residue modulo p^s. Instead of listing them consecutively,

$$0,1,2,\dots,k-1,$$

group them as

$$0, p^s, 2p^s, \dots,$$

then

$$1,1+p^s,1+2p^s,\dots,$$

and so forth.

Under this coordinate permutation, the scale-k coefficient matrices decompose into repeated blocks:

$$A_k \sim A_m \oplus A_m \oplus \cdots \oplus A_m,$$
$$B_k \sim B_m \oplus B_m \oplus \cdots \oplus B_m,$$

$$C_k \sim C_m \oplus C_m \oplus \cdots \oplus C_m,$$

with p^scopies.

The symbol ∼denotes conjugacy by a coordinate permutation.

The same linear transformation acts on every residue-class component. The scale-k block space becomes

$$\mathbb{F}_p^k \cong \left(\mathbb{F}_p^m\right)^{\oplus p^s}.$$

If the scale-m invertible context matrices generate a group

$$H_m \leq GL(m,p),$$

then the scale-k linear core acts diagonally:

$$h \cdot (v_1, \ldots, v_{p^s}) = (hv_1, \ldots, hv_{p^s}).$$

The group does not independently choose a different element of $H_m$on each residue class. The same microscopic context induces the same linear action on all copies.

The residue splitting therefore enlarges the representation space without necessarily enlarging the abstract nontranslation group.

This distinction is crucial. The total contextual group can grow rapidly because the translation module has dimension k, even while the linear quotient remains isomorphic to the group at the odd-core scale.

The larger scale contains more positions on which the group acts, but the essential non-Abelian symmetry may remain unchanged.

9.5 Contextual-group towers along characteristic-primary scale rays

Fix an integer $m$not divisible by p. Consider the sequence of scales

$$m,\ pm,\ p^2m,\ p^3m, \ldots.$$

This is a characteristic-primary scale ray.

The Frobenius decomposition implies that the contextual groups along this ray share the same nontranslation symmetry core.

For a broad class of interacting linear cellular automata, the contextual group has the form

$$\boxed{G_{p^s m} \cong \mathbb{F}_p^{p^s m} \rtimes H_m.}$$

Here

$$\mathbb{F}_p^{p^s m}$$

is the translation subgroup, and

$$H_m$$

is the linear symmetry core determined by the characteristic-free part $m$of the scale.

The sequence

$$G_m, G_{pm}, G_{p^2 m}, \ldots$$

therefore forms a Frobenius contextual-group tower.

At each stage, the translation space expands:

$$\dim_{\mathbb{F}_p} T_{p^{s+1} m} = p \dim_{\mathbb{F}_p} T_{p^s m}.$$

The quotient by translations remains fixed:

$$G_{p^s m} / T_{p^s m} \cong H_m.$$

This yields a sequence of exact extensions

$$1 \longrightarrow T_{p^s m} \longrightarrow G_{p^s m} \longrightarrow H_m \longrightarrow 1.$$

For binary rules, scale doubling gives

$$1 \longrightarrow \mathbb{F}_2^{2^s m} \longrightarrow G_{2^s m} \longrightarrow H_m \longrightarrow 1.$$

The tower is therefore not a repetition of identical groups. The translation kernel becomes larger at every stage. But the nontranslation quotient remains stable.

This is a precise form of algebraic persistence across scale.

9.6 Preservation of the nontranslation symmetry core

The nontranslation symmetry core can be defined as

$$\bar{G}_k = G_k/T_k,$$

where $T_k$is the normal subgroup of translations.

For affine contextual groups, the translation subgroup consists of maps

$$x \mapsto x + b$$

with

$$b \in \mathbb{F}_p^k.$$

The quotient

$$\bar{G}_k$$

records the linear action after translations are factored out.

If

$$k = p^s m, p \nmid m,$$

then the Frobenius theorem gives

$$\boxed{\bar{G}_{p^s m} \cong \bar{G}_m.}$$

Thus multiplying the scale by the characteristic cannot create or destroy the abstract nontranslation core in the additive setting.

If $H_m$is Abelian, then every group along the ray has an Abelian linear quotient. If $H_m$is non-Abelian, non-Abelian symmetry persists at all scales

$$p^s m.$$

If $H_m$is solvable, the linear core remains solvable. If it contains a non-Abelian simple composition factor, that factor survives along the entire ray.

The earlier empirical observation that symmetry sometimes persists under scale doubling is therefore explained by an exact theorem.

Apparent irregularity occurs primarily when comparing scales with different characteristic-free parts. For example,

$$k = 6$$

and

$$k = 8$$

have different odd parts:

$$\text{oddpart}(6) = 3,$$
$$\text{oddpart}(8) = 1.$$

They need not share the same symmetry core.

The correct arithmetic comparison is not always between adjacent scales. It is between scales lying on the same characteristic-primary ray.

9.7 Translation-module inflation under scale multiplication

Although the nontranslation core remains fixed along a Frobenius tower, the translation subgroup expands.

At scale k, a complete translation subgroup contains maps

$$\tau_b(x) = x + b$$

for all

$$b \in \mathbb{F}_p^k.$$

Its order is

$$| T_k | = p^k.$$

When the scale is multiplied by p,

$$k \mapsto pk,$$

the order becomes

$$| T_{pk} | = p^{pk}.$$

The ratio is

$$\frac{| T_{pk} |}{| T_k |} = p^{(p-1)k}.$$

For binary systems,

$$\frac{| T_{2k} |}{| T_k |} = 2^k.$$

Thus every scale doubling adds an exponentially larger family of translations.

This inflation reflects the increase in macrostate degrees of freedom. A block of length $2k$has twice as many binary coordinates as a block of length k. Even if the same linear core acts on both residue classes, translations can independently alter all $2k$coordinates.

The group order therefore grows as

$$| G_{p^s m} | = p^{p^s m} | H_m |$$

when the translation module is complete.

The translation subgroup is elementary Abelian:

$$T_k \cong (C_p)^k.$$

It is normal in the affine group, and the linear core acts on it by conjugation:

$$h\tau_b h^{-1} = \tau_{h(b)}.$$

The semidirect product records this interaction:

$$G_k = T_k \rtimes H_m.$$

The scale transition therefore has two simultaneous effects:

stable nontranslation symmetry,
expanding translational freedom.

This is a structured form of algebraic growth rather than simple replication.

9.8 Arithmetic observation scales as selectors of emergent algebra

The Frobenius tower results give observation scale a stronger role than ordinary coarse resolution.

For additive cellular automata, the algebra extracted at scale $k$depends not only on the magnitude of $k$but on its prime factorization relative to the field characteristic.

Write

$$k = p^s m.$$

The exponent $s$controls the multiplicity of residue-class copies and the dimension of the translation module. The characteristic-free part $m$controls the nontranslation symmetry core.

Thus two large scales can produce radically different groups if their characteristic-free parts differ, while widely separated scales can share the same core if they lie on one Frobenius ray.

For binary rules:

$$3,\ 6,\ 12,\ 24, \ldots$$

share odd part 3.

Likewise,

$$5,\ 10,\ 20,\ 40, \ldots$$

share odd part 5.

But powers of two all have odd part one:

$$1,\ 2,\ 4,\ 8,\ 16, \ldots.$$

They form a distinct algebraic regime.

The map

$$k \longmapsto H_{\mathrm{oddpart}(k)}$$

acts as an arithmetic phase selector.

This produces a more refined view of scale-dependent emergence:

$$\boxed{\text{microscopic rule} + \text{arithmetic type of scale} \longrightarrow \text{emergent symmetry core.}}$$

The observer does not change the local rule. The observer changes the block size. Yet that choice determines which residue decomposition, matrix structure, and finite group become available.

Scale is therefore not a passive lens. It participates mathematically in the organization of the macroalgebra.

For Algebraic Abiogenesis, this result has two implications.

First, emergent algebra need not vary continuously or monotonically with scale. It can undergo arithmetic phase transitions.

Second, the same substrate may contain an infinite family of algebraic regimes indexed by number-theoretic properties of observation.

The resulting mathematical ecology is not merely spatial or dynamical. It is arithmetic.

The Frobenius towers show that algebraic birth can be governed by prime characteristic, residue classes, and factorization of scale. Higher algebraic organization is selected not only by what the substrate does, but also by the arithmetic manner in which the substrate is assembled into observable objects.

10. Rule 60 and Rule 90 as Distinct Algebraic Phase Diagrams

10.1 The odd part of the observation scale

For binary additive cellular automata, the block size can be written uniquely as

$$k = 2^s m,$$

where $m$is odd. The integer $m$is the odd part of k:

$$m = \mathrm{oddpart}(k).$$

The Frobenius decomposition shows that the power-of-two factor and the odd factor play fundamentally different roles.

The exponent $s$determines how many residue-class copies appear after coordinates are reorganized modulo 2^s. It therefore controls the dimensional inflation of the translation module.

The odd part $m$determines the nontranslation linear symmetry core.

This separates the scale variable into two algebraically meaningful components:

$$k \leftrightarrow (2^s, m).$$

The magnitude of the block is not enough to predict its emergent symmetry. Two blocks of comparable size may have entirely different group structures because their odd parts differ.

For example,

$$k = 8$$

has odd part one, while

$$k = 9$$

has odd part nine. Although the scales differ by only one cell, they lie in radically different arithmetic regimes.

Conversely,

$$k = 3,\ 6,\ 12,\ 24$$

all share odd part three. Their block spaces grow rapidly, but their nontranslation symmetry core remains the same in the Rule 60 family.

This motivates the idea of an arithmetic phase diagram. The possible algebraic phases are indexed not by a continuous temperature or density, but by the number-theoretic decomposition of observation scale.

A phase boundary can occur whenever the odd part changes. The transition

$$k \mapsto k+1$$

may therefore cross from one group type to another even when the spatial resolution changes only slightly.

The scale parameter is not merely quantitative. It contains arithmetic structure capable of selecting qualitatively different algebras.

10.2 The Rule 60 affine-group formula

Rule 60 is the additive elementary cellular automaton with local update

$$x_i(t+1) = x_{i-1}(t) + x_i(t) (\mathrm{mod} 2).$$

Its Laurent polynomial may be written, up to orientation convention, as

$$P(z) = 1 + z.$$

At scale k, the exact supercell operation is linear in the left, central, and right input blocks. Fixing two arguments produces affine maps on

$$\mathbb{F}_2^k.$$

The complete contextual unit group has the form

$$\boxed{G_k(60) \cong \mathbb{F}_2^k \rtimes GL(m,2),}$$

where

$$m = \mathrm{oddpart}(k).$$

The general linear group acts identically on the 2^s residue-class copies of its natural m-dimensional module.

The order of the group is

$$\boxed{| G_k(60) | = 2^k \prod_{i=0}^{m-1} \left(2^m - 2^i\right).}$$

The first factor,

$$2^k,$$

is the order of the translation subgroup.

The product

$$\prod_{i=0}^{m-1}\left(2^m - 2^i\right)$$

is the order of

$$GL(m,2).$$

The formula gives an exact all-scale description rather than a finite list of computed group orders.

For example, at scale three,

$$m = 3,$$

so

$$G_3(60) \cong \mathbb{F}_2^3 \rtimes GL(3,2) = AGL(3,2).$$

Its order is

$$8 \cdot 168 = 1344.$$

At scale six,

$$6 = 2 \cdot 3,$$

so the same linear core appears:

$$G_6(60) \cong \mathbb{F}_2^6 \rtimes GL(3,2).$$

Its order is

$$64 \cdot 168 = 10{,}752.$$

The translation module has expanded from dimension three to dimension six, but the quotient by translations remains

$$GL(3,2).$$

At scale five,

$$G_5(60) \cong AGL(5,2),$$

whose order is much larger because the odd-core dimension is five.

Rule 60 therefore generates an infinite family of affine groups whose linear part is selected exactly by the odd part of block size.

10.3 Abelian behavior at power-of-two scales

When the block size is a power of two,

$$k = 2^s,$$

its odd part is

$$m = 1.$$

Since

$$GL(1,2)$$

is the trivial group, the Rule 60 formula reduces to

$$\boxed{G_{2^s}(60) \cong \mathbb{F}_2^{2^s}.}$$

The contextual unit group consists only of translations.

Every element has the form

$$x \mapsto x + b,$$

where

$$b \in \mathbb{F}_2^{2^s}.$$

Composition is vector addition:

$$\tau_a \circ \tau_b = \tau_{a+b}.$$

Because addition in $\mathbb{F}_2^{2^s}$ is commutative,

$$\tau_a \circ \tau_b = \tau_b \circ \tau_a.$$

The group is therefore Abelian.

Every nonidentity element has order two:

$$\tau_b^2 = \mathrm{id}.$$

Thus

$$G_{2^s}(60) \cong (C_2)^{2^s}.$$

This gives an infinite sequence of scales at which a rule capable of generating large non-Abelian groups at nearby scales becomes purely translational.

Examples include

$$k = 1,\ 2,\ 4,\ 8,\ 16, \ldots.$$

The group orders are

$$2,\ 4,\ 16,\ 256,\ 65{,}536, \ldots,$$

but increasing order does not imply increasing noncommutative complexity. These are all elementary Abelian two-groups.

The phenomenon illustrates why group order is an inadequate measure of algebraic richness. The scale-eight group has 256 elements but remains structurally simple. At scale seven, the

corresponding Rule 60 group contains the enormous non-Abelian group $GL(7,2)$as its linear core.

The transition

$$k = 7 \longrightarrow 8$$

therefore produces a dramatic symmetry collapse even as the block space doubles in size.

At power-of-two scales, the Frobenius identity is maximally sparse:

$$(1 + z)^{2^s} = 1 + z^{2^s}.$$

All intermediate binomial coefficients vanish modulo two. The local influence splits into isolated residue-class channels, and the nontranslation mixing disappears.

The algebraic simplicity of power-of-two scales is therefore a direct consequence of characteristic-two arithmetic.

10.4 Full affine symmetry at odd scales

When $k$is odd,

$$m = k.$$

The Rule 60 group formula becomes

$$\boxed{G_k(60) \cong \mathbb{F}_2^k \rtimes GL(k,2) = AGL(k,2).}$$

Thus every odd scale supports the full affine general linear group on the k-dimensional binary vector space.

An affine transformation has the form

$$x \mapsto Ax + b,$$

where

$$A \in GL(k,2)$$

and

$$b \in \mathbb{F}_2^k.$$

The group therefore contains every invertible linear transformation and every translation.

This is the largest possible affine permutation group on the block space compatible with its vector-space structure.

At odd scales, Rule 60 can contextually generate:

- every change of basis;
- every invertible linear mixing of block coordinates;
- every coordinate permutation;
- every elementary transvection;
- every translation;
- and every composition of these transformations.

The action is transitive because translations can send any vector to any other vector.

It is more strongly 2-transitive. Given two ordered pairs of distinct vectors

$$(x_1, x_2)$$

and

$$(y_1, y_2),$$

there exists an affine transformation sending

$$x_1 \mapsto y_1$$

and

$$x_2 \mapsto y_2.$$

To see this, translate x_1 and y_1 to zero, then choose a linear transformation sending the nonzero vector

$$x_2 - x_1$$

to

$$y_2 - y_1.$$

The scale-k symmetry is therefore highly homogeneous.

At odd scales, no coordinate direction is permanently distinguished. The group can transform any nonzero difference vector into any other.

This stands in sharp contrast with the power-of-two scales, where only translations survive and differences between vectors remain fixed.

Rule 60 therefore alternates between two qualitatively distinct regimes:

$$\begin{aligned} k \text{ odd:} &\quad \text{full affine mixing,} \\ k = 2^s\text{:} &\quad \text{pure translation.} \end{aligned}$$

Even scales with nontrivial odd parts occupy intermediate regimes in which the same general linear group acts diagonally on repeated residue-class copies.

10.5 Solvability and non-Abelian phase transitions

The group $GL(m, 2)$changes qualitatively as $m$increases.

For

$$m = 1,$$

the group is trivial.

For

$$\begin{aligned} &m = 2, \\ &GL(2,2) \cong S_3, \end{aligned}$$

which is non-Abelian but solvable.

For

$$\begin{aligned} &m = 3, \\ &GL(3,2) \end{aligned}$$

has order 168 and is isomorphic to the simple group

$$PSL(2,7).$$

It is non-Abelian and nonsolvable.

Since the odd part $m$is always odd in the Rule 60 decomposition, the first relevant values are

$$m = 1,\ 3,\ 5,\ \ldots.$$

Thus the symmetry phases include:

$$\begin{array}{ll} m = 1: & \text{Abelian,} \\ m \geq 3: & \text{non-Abelian and nonsolvable.} \end{array}$$

The first nonsolvable phase appears whenever the observation scale has odd part three:

$$k = 3 \cdot 2^{s}.$$

This includes

$$3,\ 6,\ 12,\ 24,\ \ldots.$$

Along this entire dyadic ray, the quotient by translations contains

$$GL(3,2).$$

At scales whose odd part is five, the linear core is

$$GL(5,2),$$

which is much larger and also contains non-Abelian simple composition factors.

The transition from a power-of-two scale to a nearby scale with odd part at least three can therefore create nonsolvable symmetry.

For example,

$$G_8(60) \cong (C_2)^8$$

is Abelian, while

$$G_9(60) \cong AGL(9,2)$$

has an enormous nonsolvable linear component.

The phase transition is exact and arithmetic. No parameter in the microscopic rule changes. Only the observation scale changes.

One may define the solvability phase function

$$\sigma_{60}(k) = \begin{cases} 0, & G_k(60) \text{ solvable,} \\ 1, & G_k(60) \text{ nonsolvable.} \end{cases}$$

For Rule 60,

$$\sigma_{60}(k) = 1$$

exactly when

$$\mathrm{oddpart}(k) \geq 3.$$

Since every odd part larger than one is at least three, this means that nonsolvable symmetry occurs precisely when $k$is not a power of two.

The powers of two form isolated solvable valleys inside an otherwise nonsolvable scale spectrum.

10.6 The Rule 90 Jordan-pair decomposition

Rule 90 is another additive elementary cellular automaton, with local update

$$x_i(t+1) = x_{i-1}(t) + x_{i+1}(t) (\mathrm{mod} 2).$$

Its Laurent polynomial is

$$P(z) = z^{-1} + z.$$

Unlike Rule 60, Rule 90 does not include the central cell directly. Its left-right symmetry produces a different block-matrix structure.

Let

$$k = 2^s m, m \text{ odd.}$$

Define

$$a = \frac{m+1}{2}, b = \frac{m-1}{2}.$$

The odd-core coordinate space decomposes into two subspaces of dimensions $a$and b. The linear contextual action is generated by pairs of regular unipotent Jordan transformations on these subspaces.

Let

$$J_n = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}$$

be a regular unipotent Jordan block over $\mathbb{F}_2$.

Define

$$H_n = \langle J_n, J_n^{\mathsf{T}} \rangle.$$

Then the Rule 90 contextual unit group has the form

$$\boxed{G_k(90) \cong \mathbb{F}_2^k \rtimes (H_a \times H_b).}$$

The two factors reflect the decomposition created by Rule 90's bilateral propagation.

For most dimensions,

$$H_n \cong GL(n, 2).$$

An exceptional case occurs in dimension four, where the Jordan-pair generated subgroup has a special finite-group structure rather than the full $GL(4,2)$.

The Rule 90 spectrum is therefore governed not by one general linear group of dimension m, but by a product of two groups whose dimensions sum to m:

$$a + b = m.$$

This produces a different arithmetic phase diagram even though both Rule 60 and Rule 90 are binary additive cellular automata.

10.7 Product symmetry cores and exceptional dimensions

The Rule 90 linear core is

$$H_a \times H_b.$$

This direct-product structure has several consequences.

First, the action is reducible. The odd-core vector space decomposes into two invariant components. A contextual transformation generated by the rule cannot arbitrarily mix the two components in the same manner as the full group $GL(m, 2)$.

Second, the group structure depends separately on

$$a = \frac{m+1}{2}$$

and

$$b = \frac{m-1}{2}.$$

A change in odd part can enlarge one factor, the other factor, or both.

Third, the onset of nonsolvability is delayed relative to Rule 60.

For

$$m = 1,$$

the dimensions are

$$a = 1, b = 0,$$

so the linear core is trivial.

For

$$m = 3,$$
$$a = 2, b = 1.$$

The nontrivial factor is

$$H_2 \cong GL(2,2) \cong S_3,$$

which is solvable.

For

$$m = 5,$$
$$a = 3, b = 2.$$

The factor

$$H_3 \cong GL(3,2)$$

is nonsolvable. Thus Rule 90 first enters a nonsolvable phase when the odd part reaches five.

This differs from Rule 60, which becomes nonsolvable at odd part three.

Exceptional dimensions introduce further structure. When one of $a$or $b$equals four, the corresponding Jordan-pair group is not described by the generic full-general-linear pattern. Such exceptions create localized irregularities inside the otherwise systematic arithmetic spectrum.

The exceptional case is not a failure of the theory. It shows that the scale spectrum can detect subtle finite-group generation phenomena.

A microscopic difference between including the central cell, as in Rule 60, and excluding it, as in Rule 90, propagates into a different decomposition of the macrostate symmetry group.

The block-scale algebra therefore preserves detailed information about the geometry of the local rule.

10.8 Comparison of Rule 60 and Rule 90

Rules 60 and 90 share several features.

Both are linear over

$$\mathbb{F}_2.$$

Both satisfy Frobenius scale decomposition.

Both produce contextual groups with full translation subgroups:

$$\mathbb{F}_2^k.$$

Both possess nontranslation symmetry cores determined by the odd part of block size.

Both generate infinite arithmetic towers along dyadic scale rays.

Yet their symmetry cores differ fundamentally.

For Rule 60,

$$\bar{G}_k(60) \cong GL(m, 2).$$

For Rule 90,

$$\bar{G}_k(90) \cong H_{(m+1)/2} \times H_{(m-1)/2}.$$

Rule 60 generates one irreducible or broadly mixing linear core on the odd-coordinate module.

Rule 90 splits the odd-core action into two invariant factors.

This difference affects transitivity. At odd scales, Rule 60 generates the full affine group and therefore acts 2-transitively on the block space.

Rule 90's product decomposition generally preserves additional structure and does not produce the same degree of affine homogeneity.

The solvability thresholds differ:

$$\begin{aligned} &\text{Rule 60:} \quad m \geq 3, \\ &\text{Rule 90:} \quad m \geq 5. \end{aligned}$$

Their group orders also grow at different rates.

For example, at scale three,

$$| G_3(60) | = 1344,$$

while

$$| G_3(90) | = 48.$$

At scale five,

$$G_5(60) \cong AGL(5,2),$$

while Rule 90 has a product core based on dimensions three and two.

The comparison shows that “additive cellular automaton” is not one algebraic universality class. Shared linearity provides a general framework, but the local polynomial determines the detailed macro-symmetry phase diagram.

The appropriate classification should therefore combine:

field characteristic,
local Laurent polynomial,
factorization of scale,

and

contextual group spectrum.

Rules 60 and 90 demonstrate two distinct families within the same microscopic algebraic category.

10.9 Weak Algebraic Abiogenesis in additive cellular automata

The Rule 60 and Rule 90 results require a careful interpretation within Algebraic Abiogenesis.

Their microscopic rules are already linear. Vector addition over

$$\mathbb{F}_2$$

is not an emergent feature. The block states naturally form vector spaces, and the supercell operations are induced by linear evolution.

It would therefore be misleading to claim that affine or linear algebra has arisen from a substrate lacking algebra.

The higher structures nevertheless satisfy the criteria for weak Algebraic Abiogenesis.

The local rule does not explicitly present:

$$GL(m, 2),$$
$$AGL(k, 2),$$

$$H_a \times H_b,$$

or the Frobenius tower of semidirect products.

These groups appear only after:

1. cells are aggregated into blocks;
2. the system is evolved for a scale-dependent time;
3. one-variable contexts are extracted;
4. those contexts are closed under composition;
5. invertible transformations are isolated;
6. the result is factored into translation and linear components.

The resulting groups are higher organizations of inherited microscopic linearity.

This is algebraic morphogenesis:

local field addition ⟶ scale-selected finite symmetry groups.

The observation scale determines which organization appears. A power-of-two scale yields an elementary Abelian translation group. An odd scale in Rule 60 yields a full affine general linear group. Intermediate scales yield repeated-module semidirect products.

The emergent structure is therefore not merely larger. It is qualitatively reorganized.

The additive cases also provide an explanatory theory for patterns discovered computationally in the broader CA atlas. They show that scale-dependent algebra can be governed by exact arithmetic mechanisms rather than accidental finite behavior.

Within the overall program, the nonlinear and additive examples play complementary roles.

Rule 178 demonstrates candidate strong Algebraic Abiogenesis because a semilattice law appears without a corresponding commutative microscopic binary minor.

Rule 30 demonstrates contextual operation-space formation because a complete transformation monoid emerges from controlled block contexts.

Rules 60 and 90 demonstrate weak Algebraic Abiogenesis because inherited linearity develops into exact arithmetic towers of groups.

Together, they show that algebraic birth is not one phenomenon. It ranges from the formation of new quotient laws to the compositional amplification of transformations and the scale-dependent morphogenesis of pre-existing algebraic ingredients.

11. General Theory, Novelty, and Philosophical Limits

11.1 Algebraic ecologies indexed by observation policy

The computational results suggest that a cellular automaton should not be assigned one definitive emergent algebra. The same microscopic rule can support several nonisomorphic higher structures depending on the block scale, temporal depth, quotient relation, allowed contexts, and composition policy.

The correct object is therefore an algebraic ecology

$$\mathfrak{E}(F) = \{\mathcal{A}(F, \pi) : \pi \in \Pi_F\},$$

where F is the underlying dynamical substrate and Π_F is a specified family of admissible observation policies.

An element of this ecology may be:

an exact quotient algebra,
a term-derived semilattice,
a contextual transformation monoid,
a permutation group,
a congruence lattice,
a particle-reaction presentation,
a conservation group,
or a process category.

These structures are related but not interchangeable. A quotient algebra concerns identified block states. A contextual monoid concerns transformations generated by fixing environmental variables. A conservation group is an Abelian invariant of reactions. A process category retains ordering and composition that the conservation group forgets.

The ecology perspective therefore replaces the question

What algebra does F possess?

with the more precise question

Which exact algebras become available from F under which admissible policies?

This formulation also permits comparison between substrates. Two cellular automata may have different local rules yet support isomorphic quotient algebras or contextual groups. Conversely, two rules may agree under one policy and diverge under another.

An algebraic ecology is therefore both substrate-dependent and policy-indexed. It represents the total landscape of exact mathematical organizations that a lower-level dynamics can support under a defined class of observations.

11.2 Scale as an algebraically causal variable

Scale is often treated as a passive parameter that determines how finely a system is viewed. The present results show that it can play a stronger role.

Changing the block size changes:

the macrostate space,
the induced supercell operation,
the available contexts,
the congruence lattice,
the generated monoid,
and the contextual unit group.

The observation scale therefore participates in the determination of the algebra.

For Rule 178, the semilattice quotient is unavailable at block sizes one and two but becomes exact at block size three. For Rule 30, the scale-two contextual monoid becomes the full transformation monoid T_4. For Rule 60, powers of two yield Abelian translation groups, whereas odd scales yield full affine groups.

These are not merely changes in descriptive convenience. They are changes in exact algebraic structure.

One may therefore write

$$k \longmapsto \mathcal{A}_k(F)$$

as a theory-valued function of scale.

In the additive cases, the dependence is arithmetically discontinuous. The group is selected by

$$\mathrm{oddpart}(k)$$

rather than by the magnitude of $k$alone. Two adjacent scales can belong to radically different algebraic regimes, while widely separated scales on one dyadic ray share the same nontranslation symmetry core.

Scale is thus algebraically causal in the following restricted sense: once the microscopic substrate and extraction method are fixed, changing scale changes which exact higher algebra is realized.

This does not mean that scale exerts a physical force. It means that the mathematical construction depends essentially on the scale parameter. The emergent algebra is not a property of the local rule alone.

11.3 Latent operational capacity and explicit mathematical organization

Every exact macro-level result is logically implied by the microscopic rule together with the observation policy. The higher algebra cannot contradict the lower dynamics from which it is derived.

Nevertheless, logical implication does not mean explicit availability.

A local lookup table may contain the capacity to support a semilattice quotient without containing a semilattice operation in its microscopic term language. It may support the full transformation monoid on blocks without listing those transformations. It may generate affine groups whose elements and relations are nowhere named in the local rule.

This motivates a distinction between latent operational capacity and explicit algebraic organization.

A structure is latent when its existence follows from the substrate but has not yet been formed as a stable object of mathematical treatment.

It becomes explicit when:

a macrostate space has been defined,
operations or transformations have been extracted,
closure has been established,
relations have been certified,
and the resulting structure has been presented algebraically.

The transition may be written

$$\boxed{\text{latent operational capacity} \longrightarrow \text{explicit algebraic organization.}}$$

This is the central meaning of birth in Algebraic Abiogenesis.

The word does not imply that the higher algebra was logically impossible before observation. It means that the structure was not available as an organized theory at the lower descriptive level.

This resembles the distinction between a physical system having the capacity to compute and an explicit computation being organized within it. A universal substrate may contain many possible computations, but a particular computation requires inputs, encodings, controls, and readout conventions.

Likewise, a dynamical substrate may contain many possible algebras, but each becomes explicit only under a policy that forms its objects and admissible transformations.

11.4 Comparison with coarse-graining, renormalization, and emergent computation

Algebraic Abiogenesis overlaps with several established research traditions.

Coarse-graining replaces microscopic states with macroclasses. Exact coarse-graining of cellular automata seeks partitions for which the macrostate evolution is deterministic and closed.

Renormalization studies how effective descriptions change under transformations of scale. Fixed points, cycles, and universality classes describe systems whose large-scale behavior shares common structure.

Emergent computation studies how coherent structures, particles, domains, and interactions can perform information processing not obvious from the local rule.

The present framework uses ideas from all three areas but asks a distinct question:

> Which exact algebraic signatures and theories are created by the scale transformation?

A macro-CA may be simpler than the original without generating a new algebraic law. A renormalization flow may classify dynamical behavior without computing the transformation monoid of the supercell algebra. A particle collision system may perform computation without its universal conservation group or process presentation being extracted.

Algebraic Abiogenesis therefore adds an explicit universal-algebraic layer.

The relevant outputs include:

congruences,
quotient operations,
term identities,
translation monoids,
unit groups,
semidirect-product decompositions,
and law-support spectra across scale.

The novelty does not lie in replacing coarse-graining or renormalization. It lies in treating changes of scale and policy as mechanisms that can generate new exact algebraic organizations.

11.5 Comparison with universal algebra and translation monoids

Universal algebra studies sets equipped with operations of specified arities. It provides the concepts of terms, identities, congruences, quotients, subalgebras, clones, and varieties.

Translation monoids arise by fixing all but one argument of fundamental operations and closing the resulting unary maps under composition.

These tools are not new. The contextual transformation monoid

$$\mathsf{C}_k(F)$$

is naturally the translation monoid of the scale-k block algebra

$$\mathcal{B}_k(F).$$

The contribution of the present framework is not the invention of this algebraic machinery. It is the use of that machinery in a two-stage generative construction:

microscopic dynamics ⟶ finite block algebra ⟶ translation monoid and quotient theories.

Traditional universal algebra usually begins after the signature has been supplied. Here the finite algebra itself is generated from spacetime dynamics.

The framework also studies variation across an infinite family of algebras:

$$\mathcal{B}_1(F), \mathcal{B}_2(F), \mathcal{B}_3(F), \ldots.$$

This turns universal-algebraic invariants into scale spectra.

For example,

$$\mathrm{Con}\,\mathcal{B}_k(F)$$

records the exact observation policies available at scale k, while

$$\mathrm{Trans}\,\mathcal{B}_k(F)$$

records its contextual transformation capacity.

The resulting research program can be described as dynamical universal algebra: finite algebras are generated from local dynamics and then compared across scale, policy, and substrate.

11.6 What is established prior art

Several broad ingredients of this work have substantial precedent and should not be claimed as original.

Cellular automata have long been studied using algebra, topology, symbolic dynamics, category theory, finite fields, and polynomial representations.

Exact block transformations and coarse-grained cellular automata are established.

Linear cellular automata have well-known Laurent-polynomial representations and characteristic-dependent self-similarity.

Universal algebra has previously been applied to cellular automata.

Transformation monoids and translation monoids are standard finite-algebraic objects.

Rule 30, Rule 60, Rule 90, Rule 110, and Rule 178 have extensive prior literatures.

NAND is a familiar functionally complete Boolean operation.

The Frobenius identity

$$P(z)^{p^s} = P(z^{p^s})$$

is classical.

General linear groups, affine groups, wreath products, and Jordan-generated groups are established mathematical structures.

The philosophical idea that higher organization can emerge from lower-level dynamics is also not new.

The present work should therefore avoid broad claims such as:

the first algebra derived from a cellular automaton,
the first use of groups in cellular automata,

or

the first mathematical theory of emergence.

Such claims would be false or indefensible.

A credible novelty statement must concern the specific integration, exact constructions, classifications, and theorems developed here.

11.7 What appears novel in the present framework

The strongest novelty candidates lie in five areas.

First, Algebraic Abiogenesis is formulated as a signature-forming process rather than only a theorem-discovery process. The system is permitted to discover the type of algebra, not merely laws inside a supplied structure.

Second, the framework explicitly indexes emergent algebra by an observation policy:

$$(F, \pi) \mapsto \mathcal{A}(F, \pi).$$

This joins admissibility, quotient formation, context selection, and scale into one construction.

Third, the notions of algebraic genesis, persistence, extinction, and rebirth provide a formal language for tracking equational and group-theoretic properties across scales.

Fourth, the specific computational results appear unreported:

the exact Rule 178 semilattice threshold,
$\mathrm{C}_2(30) = T_4$,
the complete fourteen-rule T_4 classification,
the scale-two and scale-three contextual group taxonomy,
the Rule 60 odd-part group formula,
and the Rule 90 Jordan-pair decomposition.

Fifth, the Frobenius contextual-group tower appears to provide a new general relation between characteristic-primary scale multiplication and preservation of the nontranslation symmetry core.

The integrated novelty can be summarized as

a computational and formal theory of exact algebraic structures becoming available under changes of dynamic

These are strong novelty candidates, not guarantees of absolute historical priority. Independent reproduction, specialist review, and a comprehensive reference analysis remain necessary.

11.8 Why cellular automata are not nonmathematical substrates

A cellular automaton already contains extensive mathematical structure.

It has:

a set of states,
an equality relation,
an ordered lattice,
a neighborhood geometry,
discrete time,
and a deterministic function.

The computer executing it contains still more structure: Boolean logic, memory addressing, arithmetic, sequencing, and symbolic representation.

It would therefore be incoherent to claim that cellular automata generate mathematics from absolute non-mathematics.

Even the identification of an output as a group, semilattice, or monoid depends on pre-existing mathematical concepts.

The appropriate comparison is not

non-mathematics ⟶ mathematics.

It is

lower mathematical organization ⟶ higher mathematical organization.

The lower substrate may lack the higher structure in its native vocabulary. A two-state local rule does not possess S_4as a microscopic permutation group. A projection-like binary minor does not possess a commutative semilattice law. Yet the dynamical system can support those structures after objects and transformations are reorganized.

This is analogous to biological emergence. Chemistry is not nonphysical, and prebiotic matter is not structureless. Life arises from already lawful matter through a new level of organization.

The analogy should not be pushed too far, but it clarifies the intended meaning of abiogenesis.

11.9 The limits of the phrase mathematics from nothing

The phrase mathematics from nothing is rhetorically powerful but philosophically misleading.

There is no literal nothing in the construction. The state alphabet, transition rule, spacetime geometry, computer implementation, search algorithm, and human interpretation are all already present.

Even an anonymous operation table presupposes objects, inputs, outputs, and equality.

The phrase also risks confusing three different claims:

mathematics was not named in advance,
mathematics was not explicitly represented microscopically,
mathematics did not logically follow from the substrate.

The first two can be true. The third cannot be true for an exact deterministic derivation.

If a semilattice is exactly derived from Rule 178, then its existence is a consequence of Rule 178 and the quotient policy. If $T_4$is generated by Rule 30 contexts, then that closure was always mathematically implied by the finite tables.

The novelty lies not in escaping implication but in discovering and organizing what the implication contains.

A more accurate phrase is:

mathematics before naming

or

higher algebra from lower operational structure.

The original phrase may remain useful as a provocative question, but it should not appear as the final technical claim.

11.10 Algebraic Abiogenesis as a precise form of mathematical emergence

The cumulative results support the following definition:

Algebraic Abiogenesis is the autonomous formation and certification of algebraic objects, operations, transformations, and laws from lower-level dynamics under explicit observation policies, without specifying the target algebraic structure in advance.

This definition contains several essential restrictions.

The formation must be autonomous in the limited computational sense that the target law is not inserted into the search.

The resulting structure must be exact or clearly identified as approximate.

The observation policy must be explicit.

The microscopic comparison class must be stated when noninheritance is claimed.

The result must be reproducible and certifiable.

The phrase algebraic birth refers to the first policy or scale at which the structure becomes available as an exact algebraic object.

Under this definition, Rule 178 provides an example of law birth, Rule 30 provides an example of transformation-space birth, and Rules 60 and 90 provide examples of arithmetic algebraic morphogenesis.

The broader theory can be summarized by

$$\boxed{(F,\pi) \longmapsto \langle O_{F,\pi}, \Sigma_{F,\pi} \mid R_{F,\pi} \rangle.}$$

The substrate and policy jointly produce:

$O_{F,\pi}$:formed objects,
$\Sigma_{F,\pi}$:formed operations or transformations,
$R_{F,\pi}$:certified relations.

This is neither mathematics from literal nothing nor merely the application of known algebra to a known rule. It is a formal study of the transition by which latent operational possibilities become explicit mathematical theories.

That transition is the defensible core of Algebraic Abiogenesis.

13. References

1. von Neumann, John. Theory of Self-Reproducing Automata. Edited and completed by Arthur W. Burks. Urbana: University of Illinois Press, 1966.
2. Hedlund, Gustav A. “Endomorphisms and Automorphisms of the Shift Dynamical System.” Mathematical Systems Theory 3 (1969): 320–375. DOI: 10.1007/BF01691062.
3. Wolfram, Stephen. “Statistical Mechanics of Cellular Automata.” Reviews of Modern Physics 55, no. 3 (1983): 601–644. DOI: 10.1103/RevModPhys.55.601.
4. Wolfram, Stephen. “Universality and Complexity in Cellular Automata.” Physica D: Nonlinear Phenomena 10, nos. 1–2 (1984): 1–35. DOI: 10.1016/0167-2789(84)90245-8.
5. Wolfram, Stephen. A New Kind of Science. Champaign, Illinois: Wolfram Media, 2002.
6. Sarkar, Palash. “A Brief History of Cellular Automata.” ACM Computing Surveys 32, no. 1 (2000): 80–107. DOI: 10.1145/349194.349202.
7. Kari, Jarkko. “Theory of Cellular Automata: A Survey.” Theoretical Computer Science 334, nos. 1–3 (2005): 3–33. DOI: 10.1016/j.tcs.2004.11.021.
8. Ceccherini-Silberstein, Tullio, and Michel Coornaert. Cellular Automata and Groups. Berlin: Springer, 2010. DOI: 10.1007/978-3-642-14034-1.
9. Lind, Douglas, and Brian Marcus. An Introduction to Symbolic Dynamics and Coding. Cambridge: Cambridge University Press, 1995.
10. Martin, Olivier, Andrew M. Odlyzko, and Stephen Wolfram. “Algebraic Properties of Cellular Automata.” Communications in Mathematical Physics 93, no. 2 (1984): 219–258. DOI: 10.1007/BF01223745.
11. Pedersen, John. “Cellular Automata as Algebraic Systems.” Complex Systems 6, no. 3 (1992): 237–250.
12. Moore, Cristopher, and Arthur A. Drisko. “Algebraic Properties of the Block Transformation on Cellular Automata.” Complex Systems 10, no. 3 (1996): 185–194.
13. Israeli, Navot, and Nigel Goldenfeld. “Computational Irreducibility and the Predictability of Complex Physical Systems.” Physical Review Letters 92, no. 7 (2004): 074105. DOI: 10.1103/PhysRevLett.92.074105.
14. Israeli, Navot, and Nigel Goldenfeld. “Coarse-Graining of Cellular Automata, Emergence, and the Predictability of Complex Systems.” Physical Review E 73, no. 2 (2006): 026203. DOI: 10.1103/PhysRevE.73.026203.

15. Song, Yerim, and Joshua A. Grochow. “An Improved Algorithm for Coarse-Graining Cellular Automata.” arXiv:2012.12153, 2020.

16. Hanson, James E., and James P. Crutchfield. “Computational Mechanics of Cellular Automata: An Example.” Physica D: Nonlinear Phenomena 103, nos. 1–4 (1997): 169–189. DOI: 10.1016/S0167-2789(96)00259-X.

17. Rupe, Adam, and James P. Crutchfield. “Local Causal States and Discrete Coherent Structures.” Chaos 28, no. 3 (2018): 033101. DOI: 10.1063/1.5021130.

18. Shalizi, Cosma Rohilla, and James P. Crutchfield. “Computational Mechanics: Pattern and Prediction, Structure and Simplicity.” Journal of Statistical Physics 104 (2001): 817–879. DOI: 10.1023/A:1010388907793.

19. Crutchfield, James P. “The Calculi of Emergence: Computation, Dynamics, and Induction.” Physica D: Nonlinear Phenomena 75, nos. 1–3 (1994): 11–54. DOI: 10.1016/0167-2789(94)90273-9.

20. Cook, Matthew. “Universality in Elementary Cellular Automata.” Complex Systems 15, no. 1 (2004): 1–40. DOI: 10.25088/ComplexSystems.15.1.1.

21. Martínez, Genaro J., Andrew Adamatzky, and Harold V. McIntosh. “Complete Characterization of Structure of Rule 54.” Complex Systems 23, no. 3 (2014): 259–293. DOI: 10.25088/ComplexSystems.23.3.259.

22. Redeker, Markus. “Gliders and Ether in Rule 54.” arXiv:1007.2920, 2010.

23. Redeker, Markus. “A Language for Particle Interactions in One-Dimensional Cellular Automata.” arXiv:1012.0158, 2010.

24. Martínez, Genaro J., Andrew Adamatzky, and Harold V. McIntosh. “Phenomenology of Glider Collisions in Cellular Automaton Rule 54 and Associated Logical Gates.” Chaos, Solitons & Fractals 28, no. 1 (2006): 100–111. DOI: 10.1016/j.chaos.2005.05.013.

25. Birkhoff, Garrett. “On the Structure of Abstract Algebras.” Proceedings of the Cambridge Philosophical Society 31, no. 4 (1935): 433–454. DOI: 10.1017/S0305004100013463.

26. Burris, Stanley, and H. P. Sankappanavar. A Course in Universal Algebra. New York: Springer, 1981. Freely available Millennium Edition, 2012.

27. McKenzie, Ralph N., George F. McNulty, and Walter F. Taylor. Algebras, Lattices, Varieties, Volume I. Belmont, California: Wadsworth & Brooks/Cole, 1987.

28. Grätzer, George. Lattice Theory: Foundation. Basel: Birkhäuser, 2011. DOI: 10.1007/978-3-0348-0018-1.

29. Howie, John M. Fundamentals of Semigroup Theory. Oxford: Clarendon Press, 1995.

30. Ganyushkin, Olexandr, and Volodymyr Mazorchuk. Classical Finite Transformation Semigroups: An Introduction. London: Springer, 2009. DOI: 10.1007/978-1-84800-281-4.

31. Araújo, João, and Peter J. Cameron. “Permutation Groups and Transformation Semigroups: Results and Problems.” arXiv:1308.3585, 2013.

32. Dixon, John D., and Brian Mortimer. Permutation Groups. Graduate Texts in Mathematics 163. New York: Springer, 1996. DOI: 10.1007/978-1-4612-0731-3.

33. Cameron, Peter J. Permutation Groups. London Mathematical Society Student Texts 45. Cambridge: Cambridge University Press, 1999.

34. Yildiz, Volkan. “Translation Monoids and Recursive Evaluation in Finite Binary Algebras.” arXiv:2604.01486, 2026.

35. Buliga, Marius. “Emergent Algebras.” arXiv:0907.1520, 2009.

36. Buliga, Marius. “Origin of Emergent Algebras.” arXiv:1304.3694, 2013.

37. Martin, Olivier, Andrew M. Odlyzko, and Stephen Wolfram. “Algebraic Properties of Cellular Automata.” Communications in Mathematical Physics 93, no. 2 (1984): 219–258. DOI: 10.1007/BF01223745.

38. Willson, Stephen J. “Cellular Automata Can Generate Fractals.” Discrete Applied Mathematics 8, no. 1 (1984): 91–99. DOI: 10.1016/0166-218X(84)90082-9.

39. Wolfram, Stephen. “Geometry of Binomial Coefficients.” American Mathematical Monthly 91, no. 9 (1984): 566–571. DOI: 10.2307/2323735.

40. Allouche, Jean-Paul, Fritz von Haeseler, Heinz-Otto Peitgen, and Guentcho Skordev. “Linear Cellular Automata, Finite Automata and Pascal’s Triangle.” Discrete Applied Mathematics 66, no. 1 (1996): 1–22. DOI: 10.1016/0166-218X(94)00132-W.

41. Sutner, Klaus. “De Bruijn Graphs and Linear Cellular Automata.” Complex Systems 5, no. 1 (1991): 19–30.

42. Willson, Stephen J. “Calculating Growth Rates and Moments for Additive Cellular Automata.” Discrete Applied Mathematics 35, no. 1 (1992): 47–65.

43. Lidl, Rudolf, and Harald Niederreiter. Finite Fields. Second edition. Cambridge: Cambridge University Press, 1997.

44. Grove, Larry C. Classical Groups and Geometric Algebra. Graduate Studies in Mathematics 39. Providence: American Mathematical Society, 2002.

45. Carter, Roger W. Finite Groups of Lie Type: Conjugacy Classes and Complex Characters. Chichester: Wiley, 1985.

46. Craven, David A. “Groups with a p-Element Acting with a Single Non-Trivial Jordan Block on a Simple Module in Characteristic p.” Preprint, 2017.

47. Pellegrini, Marco A., and Alexandre E. Zalesski. “Minimal Generation of Finite Simple Groups of Lie Type by Regular Unipotent Elements.” European Journal of Mathematics, 2026. DOI: 10.1007/s40879-026-00895-4.

48. Mac Lane, Saunders. Categories for the Working Mathematician. Second edition. Graduate Texts in Mathematics 5. New York: Springer, 1998.

49. Leinster, Tom. Higher Operads, Higher Categories. London Mathematical Society Lecture Note Series 298. Cambridge: Cambridge University Press, 2004.

50. Baez, John C., and Mike Stay. “Physics, Topology, Logic and Computation: A Rosetta Stone.” In New Structures for Physics, edited by Bob Coecke, 95–172. Berlin: Springer, 2011. DOI: 10.1007/978-3-642-12821-9_2.

51. Fong, Brendan, and David I. Spivak. Seven Sketches in Compositionality: An Invitation to Applied Category Theory. Cambridge: Cambridge University Press, 2019.

52. Colton, Simon. Automated Theory Formation in Pure Mathematics. London: Springer, 2002. DOI: 10.1007/978-1-4471-0147-5.

53. Lenat, Douglas B. “AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search.” Ph.D. dissertation, Stanford University, 1976.

54. Lenat, Douglas B. “The Nature of Heuristics.” Artificial Intelligence 19, no. 2 (1982): 189–249. DOI: 10.1016/0004-3702(82)90007-X.

55. Fajtlowicz, Siemion. “On Conjectures of Graffiti.” Discrete Mathematics 72, nos. 1–3 (1988): 113–118. DOI: 10.1016/0012-365X(88)90204-9.

56. Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, et al. “Array Programming with NumPy.” Nature 585 (2020): 357–362. DOI: 10.1038/s41586-020-2649-2.

57. Meurer, Aaron, Christopher P. Smith, Mateusz Paprocki, et al. “SymPy: Symbolic Computing in Python.” PeerJ Computer Science 3 (2017): e103. DOI: 10.7717/peerj-cs.103.

58. Youvan, Douglas C. Mathematical It from Bit: Algebraic Abiogenesis, Anonymous Finite Operations, and the Automated Birth of Abstract Algebra. July 2026. DOI: 10.13140/RG.2.2.25987.26403.

59. Youvan, Douglas C. No-Corpus Artificial Intelligence from Algebraic Abiogenesis: Policy-Governed Object Formation, Theory Ecologies, and Certified Memory. July 2026. DOI: 10.13140/RG.2.2.12146.06088.

60. Youvan, Douglas C. Object Formation and Conceptual Resolution: Stable Quotients, Complete Refinement Lattices, and Functorial Observation. July 2026. DOI: 10.13140/RG.2.2.25921.72800.

61. Youvan, Douglas C. How Mathematics Makes Objects: Observation, Identity, and a New Algebra of Concept Formation. July 2026. DOI: 10.13140/RG.2.2.13338.81603.

62. Youvan, Douglas C. The Mathematics of Concept Formation: Discovery Calculus, Explanation Geometry, Theory Dynamics, and the Formal Birth of New Objects. July 2026. DOI: 10.13140/RG.2.2.23562.43209.

63. Youvan, Douglas C. Cellular Algebraic Abiogenesis: Compact Computational Release. Version 0.6.1. GitHub software repository, July 2026. MIT License.

The author has published over 5,700 AI-assisted papers at: https://www.researchgate.net/profile/Douglas-Youvan/research . Google Scholar has indexed these preprints, and if you want a particular reference, the AI mode of a Google search (Gemini) has efficiently catalogued these preprints.